UNIVERSIDAD CARLOS III DE MADRID

PROYECTO FIN DE GRADO

DESARROLLO DE LIBRERÍA DE ESTIMACIÓN DE DISTANCIA PARA APLICACIÓN MÓVIL. APLICACIÓN A PEATONES

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

Autor: Victor Cancho Armesto

Tutor: Fernando García Fernández

Leganés, 24 de septiembre de 2013

Índice

Índice de Figuras

1. Introducción

1.1 Motivación

La ayuda en la conducción es un tema de gran actualidad. Gracias al gran desarrollo de las tecnologías y la infinidad de posibilidades que estas nos proporcionan no es descabellado pensar en una pronta implementación de cantidad de ayudas a la conducción, incluso cabe la posibilidad de que el coche actúe de forma totalmente autónoma dependiendo de las necesidades.

Este interés en el desarrollo de las ayudas a la conducción es sin duda propiciado por la gran cantidad de accidentes que se producen a diario por el uso de automóviles. El objetivo de este proyecto es por tanto contribuir a mejorar estas estadísticas.

Para ello se ha trabajado con Smartphones, con el sistema operativo Android. Las razones de esta elección son múltiples, pero las más importantes son, las grandes expectativas de futuro, la gran importancia que han tomado los smartphones en los últimos años, y la gran facilidad de uso que estos dispositivos nos ofrecen.

Como se ha podido ver durante la realización del trabajo también existen grandes inconvenientes para usar los smartphones en esta línea de investigación, pero se espera que estos problemas se vayan solucionando según madure el sistema operativo Android y según los fabricantes de dispositivos móviles se den cuenta de las necesidades de los investigadores.

La primera parte de este proyecto consiste en la creación de una librería para Android la cual nos proporcione datos útiles orientados al trabajo con la cámara del dispositivo y el cálculo de distancias mediante el modelo Pin-hole. Estos datos que nos proporciona serán los datos intrínsecos de la cámara ya la orientación del dispositivo con respecto al suelo.

La segunda parte de este proyecto será la correcta aplicación de dicha librería a un software, ya existente, de detección de peatones para calcular correctamente las distancias que separan al dispositivo a dicho peatón.

Esto deja abierta la puerta a continuar el desarrollo de aplicaciones que sean capaces de dar la alarma cuando un peatón se cruce en la trayectoria de nuestro vehículo o que nos avise que el coche que circula inmediatamente delante de nosotros está frenando.

1.2 Entorno socio económico

Aunque parezca que en España se han conseguido reducir en los últimos años debido al gran despliegue de campañas de concienciación, la subida de las multas, los radares de

velocidad desplegados por toda la red de carreteras, y el carnet por puntos, los accidentes siguen siendo muchos y las victimas mortales también.

Para hacernos una idea durante el año 2013, en las vías interurbanas se han producido 994 accidentes mortales en los que han fallecidos 1.128 personas y 5.206 han resultado heridas graves. Esto significa que se ha producido un descenso del 16% en el número de accidentes, del 13,3% en el de víctimas mortales y 16% en el de heridos graves con respecto a 1960.

La DGT nos proporciona datos históricos comparando el año 1960 con el año 2013 siendo este el análisis. La cifra de fallecidos registrada en 2013 es menor a la de 1960 cuando hubo 1.300 muertos, primer año en que se tienen estadísticas. Teniendo en cuenta que el escenario de movilidad es absolutamente distinto (en 1960 había un millón de vehículos y en 2013 el parque automovilístico ascendió a 31 millones).

Entre 2005 y 2013, el número de víctimas mortales ha disminuido un 65%. En el mismo periodo, el número de movimientos de largo recorrido ha disminuido un 10%, mientras que el parque de vehículos y el censo de conductores han aumentado un 13% y un 22%, respectivamente.

Estos datos, que en cierto modo son positivos, nos dejan un sabor agridulce cuando nos fijamos en las víctimas mortales en un solo año, y nos crean la necesidad de llevar una línea de investigación dedicada a conseguir que este número de muertos se reduzca.

A pesar de que en España se haya conseguido reducir el número de accidentes, se trata de un problema a nivel global. Es más, como se puede observar en el siguiente mapa mundial, los accidentes en los países menos desarrollados son más y también las víctimas mortales.

Figura 1

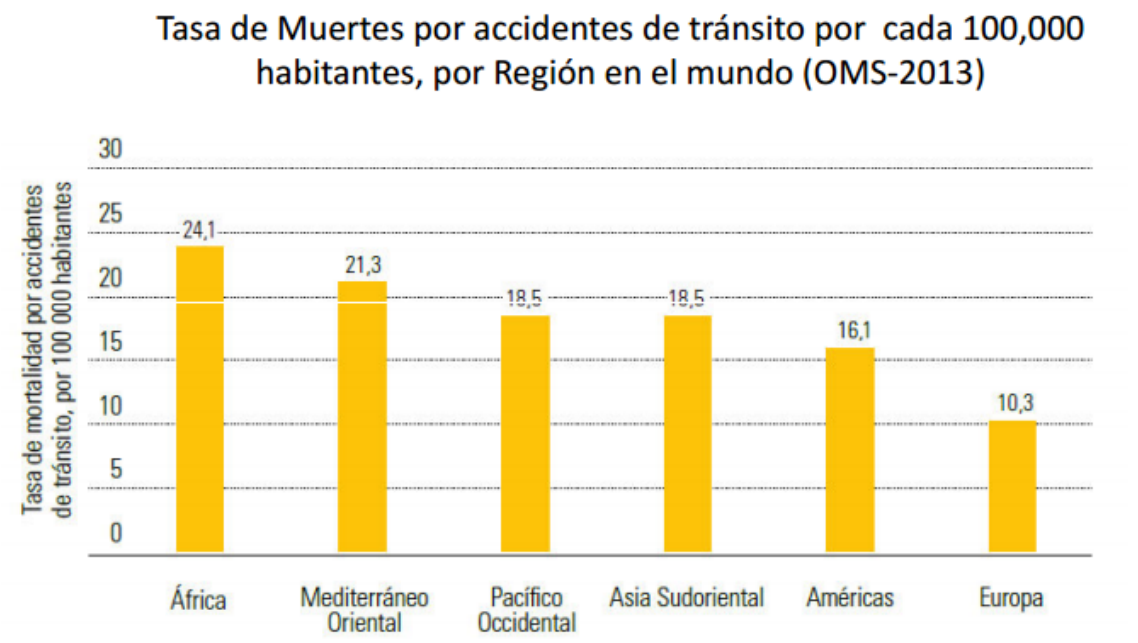


Figura 2

Todas las personas que mueren, se lesionan o quedan discapacitadas por un accidente de tráfico tienen una red de personas allegadas, como familiares y amigos, que resultan profundamente afectadas. En el mundo, millones de personas se enfrentan a la muerte o la discapacidad de familiares debido a lesiones causadas por el tráfico. Sería imposible asignar un valor cuantitativo a cada caso de sacrificio y sufrimiento humano, sumarlos todos y obtener una cifra que refleje el costo social mundial de los choques y las lesiones causadas por el tráfico.

Se estima, sin embargo, que el costo económico de los choques y las lesiones causadas por el tráfico asciende al 1% del producto nacional bruto (PNB) en los países de ingresos bajos, al 1,5% en los de ingresos medianos y al 2% en los de ingresos altos. El costo mundial se estima en US$ 518.000 millones anuales, de los cuales US$ 65.000 millones corresponden a los países de ingresos bajos y medianos.

Las lesiones causadas por el tráfico representan una pesada carga no sólo para la economía mundial y de los países, sino también para la de las familias. La pérdida de quienes ganaban el sustento y el costo añadido de atender a los familiares discapacitados por dichas lesiones sumen a muchas familias en la pobreza.

Víctimas mortales según el tipo de vehículo.

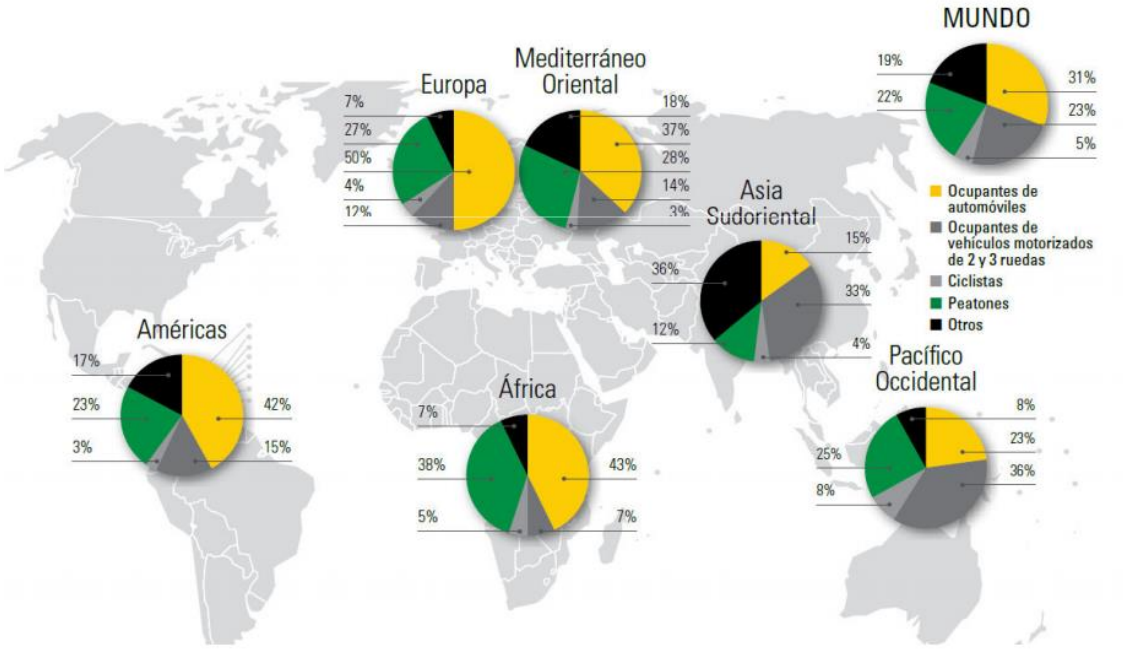


Figura 3

2. Estado del arte

Desde hace mucho tiempo ya se ha investigado en distintos dispositivos para nuestros vehículos tratando de evitar accidentes o por lo menos suavizarlos. Dichos dispositivos son llamados “dispositivos de ayuda a la conducción” y pueden ser de dos tipos, activos o pasivos.

En estos momentos estamos en el boom informático por lo que los dispositivos que están siendo desarrollados son de tipo activo ya que pretenden adelantarse a las reacciones de los conductores e incluso interactuar con ellos para así evitar los accidentes.

La mayoría de dichos dispositivos aún están en desarrollo y sin perfeccionar por lo que aún no suponen una diferencia real en la conducción del día a día ni se puede decir que salven vidas, pero parece claro que el día que se implementen en todos los coches, y este día no queda muy lejano según las previsiones, la conducción tal como la conocemos hoy en día cambiará drásticamente.

Más tarde se comentaran las distintas ayudas a la conducción que se están desarrollando actualmente y cuáles de ellas ya han sido implementadas por algunas marcas de vehículos, pero antes vamos a centrarnos en los dispositivos pasivos que tantas vidas han salvado.

Figura 4

Los dispositivos de seguridad pasivos se desarrollaron, en su mayoría, a lo largo de la segunda mitad del siglo XX. La necesidad de ellos surgió debido al gran avance en los motores de los vehículos y la potencia que eran capaces de desarrollar. De repente existían vehículos capaces de acelerar muy deprisa y mantener una velocidad punta muy alta, y en cambio no tenían ningún tipo de seguridad en caso de colisión.

2.1 Sistemas pasivos

Los sistemas pasivos son elementos que tratan de reducir al mínimo los daños que se puedan producir en caso de accidente.

2.1.1 Cinturón de seguridad

Figura 5

Posiblemente es el sistema de seguridad pasiva más importante de todos, y el que teóricamente ha salvado más vidas desde su invención. El cinturón de seguridad es obligatorio en nuestro país desde 1974 (para carretera) y desde 1992 en entorno urbano. Utilizado correctamente, el cinturón reduce en nueve veces las probabilidades de fallecimiento en caso de colisión, y reduce sensiblemente las posibilidades de sufrir lesiones de cuello y cabeza, entre otras. Evitan que los ocupantes salgan despedidos.

2.1.2 Pretensor y limitador del cinturón

Los limitadores son elementos mecánicos que bloquean el cinturón, de forma que no "dé de sí".

El pretensor pirotécnico lo que hace es, una vez detectada la colisión por la centralita, tirar del cinturón para apretarlo contra el cuerpo. En algunos coches de alta gama, incluso, el sistema realiza un ajuste previo del cinturón tensándolo de forma notoria contra el cuerpo, para calcular así de forma perfecta la fuerza que deberá hacer para tensar la cinta en caso de choque. Así, se maximiza el efecto salvavidas del cinturón de seguridad.

2.1.3 Airbag

Figura 6

Son unas bolsas que, mediante un sistema pirotécnico, se inflan en fracciones de segundo cuando el coche choca con un objeto sólido a una velocidad considerable. Su objetivo es contener la cabeza y el cuerpo, de forma que se minimiza su desplazamiento.

Actualmente existen las bolsas frontales, laterales, tipo cortina (para la cabeza) e incluso para las rodillas.

2.1.4 Chasis y carrocería

Figura 7

En ambos existen zonas que absorben la energía en caso de un impacto. Si es un choque frontal, acomoda el motor para que no se introduzca en el habitáculo.

Cuando un peatón es atropellado, las heridas que puede sufrir son muy específicas. Normalmente, toda la zona de la cadera y piernas, y también la cabeza, pueden recibir daños dependiendo de cómo se produzca el atropello. La idea es que no solo recibe el golpe del coche, sino que en los momentos inmediatos puede “rebotar” en diferentes partes del capó y parabrisas del coche.

Para minimizar en la medida de lo posible las consecuencias de un atropello se han desarrollado diversos sistemas de seguridad que permiten que la carrocería se deforme de forma específica, de modo que la energía del impacto se absorba a través la estructura antes que por el cuerpo del peatón.

2.1.5 Cristales

Figura 8

El compuesto del cristal parabrisas está preparado para que, en caso de accidente, no salten astillas que puedan dañar a los pasajeros del vehículo. Las ventanillas laterales son más débiles y se pueden romper. Es la salida más cómoda si en caso de vuelco las puertas se quedan bloqueadas.

2.1.6 Reposacabezas

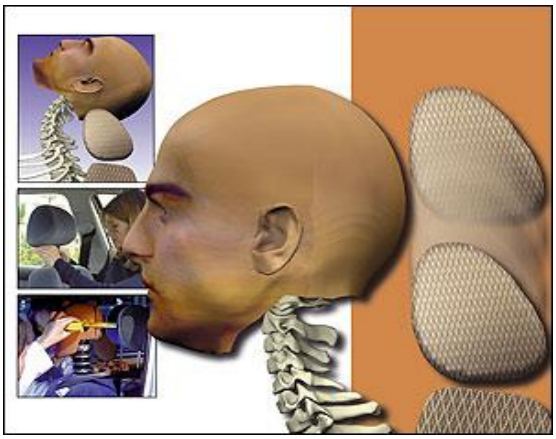


Figura 9

Son los elementos fundamentales en la protección de la persona frente al latigazo cervical, siempre que se ajusten a la altura de la persona que vaya sentada.

2.1.7 Corte de inyección (en colisión)

Este sistema ayuda a que no se produzca un incendio tras una colisión. Simplemente, detiene la inyección de combustible al motor cuando una línea de combustible se rompe, aislando el depósito y la bomba de combustible del motor. De este modo, cualquier posibilidad de incendio queda reducida al mínimo. Esto no significa que nunca se pueda producir un incendio, pero sí que se reduce mucho la posibilidad de que suceda.

2.2 Sistemas activos

Los sistemas activos son aquellos que velan por nuestra seguridad. Cada día son más y más sofisticados. Estos sistemas son aquellos que trabajan para reducir el riesgo de sufrir un accidente.

2.2.1 Sistema de frenado

Su función es fundamental para la seguridad del conductor. Todos los sistemas de frenado actuales cuentan con circuitos independientes que permiten frenar con seguridad en caso de que alguno falle. Entre los mejores se encuentran los antibloqueo (ABS) que reducen la distancia de frenado manteniendo la capacidad de cambiar de dirección para evadir obstáculos, ya que no bloquean las ruedas. El ABS detecta qué rueda está sufriendo deslizamiento y libera presión del sistema de frenos.

2.2.2 Sistema de suspensión activa

El automóvil se mantiene estable y absorbe las irregularidades de la carretera. Las barras estabilizadoras conectan las dos ruedas de cada eje y sirven para controlar la inclinación del coche en las curvas, evitando así una salida de la vía.

El sistema funciona a partir de los datos recogidos por varios conjuntos de sensores (en ruedas y amortiguadores, sensores giroscópicos) y controla los amortiguadores (con sistemas neumáticos o hidráulicos en su interior), cuya función es variar la dureza del amortiguador.

2.2.3 Sistema de dirección

Garantiza la correcta maniobra del vehículo. Los sistemas de dirección de los coches actuales se endurecen a altas velocidades para evitar posibles accidentes.

2.2.4 Control de estabilidad

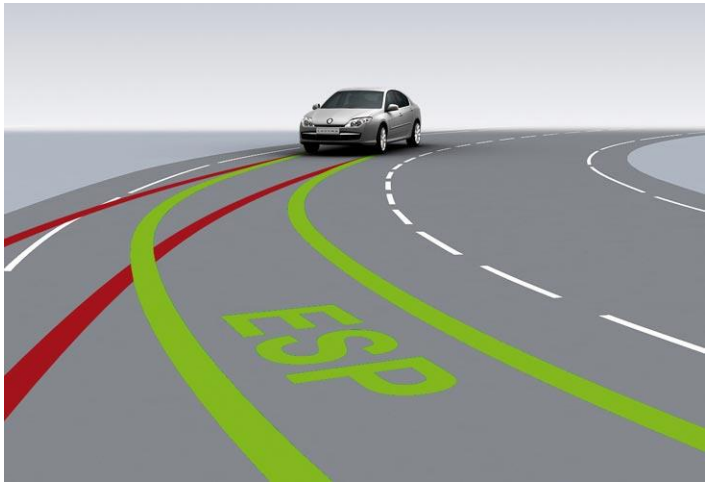


Figura 10

El control de estabilidad (ESP) trata de mantener al coche en la trayectoria correcta, actuando sobre el conjunto de las cuatro ruedas.

El ESP actúa comparando la trayectoria elegida (ángulo de dirección) con la real (ángulo de giro y dirección transversal), y contrastando las dos con los datos de velocidad de giro de cada rueda. Eso da una visión muy clara de si el conductor tiene el control del coche o no, pues normalmente tanto el ángulo de dirección como el ángulo de giro deben coincidir, y las ruedas no deben de deslizar. La función del ESP es corregir cualquier irregularidad para conseguir adaptar el ángulo de giro real al deseado.

2.2.5 Control de tracción

El control de tracción, o TCS, es un sistema de seguridad activa que trabaja para que los neumáticos de las ruedas motrices se mantengan en contacto con la calzada sin patinar. Esto puede ser, por ejemplo, arrancando en una pendiente deslizante o en cualquier situación en la que se necesite tracción.

El TCS comparte sensores con el ABS, pues necesita saber el estado de cada rueda, si empieza a patinar o no. Además, es un sistema que funciona con cada rueda de forma independiente, y por decirlo de una forma sencilla, si una rueda empieza a deslizar, simplemente la frenará de alguna manera para neutralizar ese patinazo.

2.2.6 Alerta de cambio involuntario de carril

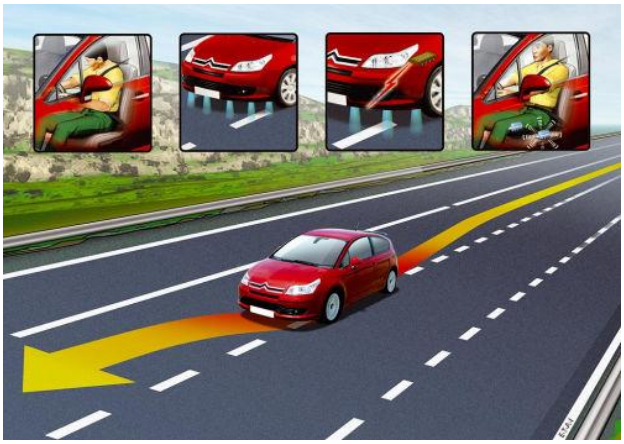


Figura 11

El sistema avisa al conductor cuando éste, de forma inadvertida, se sale de la trayectoria del carril por el que circula e invade un carril adyacente. Lo hace mediante unos sensores infrarrojos los cuales detectan la línea de la carretera y si la hemos sobrepasado.

La marca de vehículos Citroën implementa este sistema en sus nuevos vehículos y afirman que se podrían evitar el 10% de accidentes en España con este sistema.

2.2.7 Control de crucero inteligente

Figura 12

Al igual que ocurre con el Control de Velocidad de Crucero, regula la velocidad a la que queremos circular de forma automática. La novedad del ACC es que, con la ayuda de un sistema de radar controla, también de forma automática, la distancia de circulación con respecto al vehículo precedente, frenando nuestro vehículo si es necesario para mantener dicha distancia de seguridad.

2.2.8 Asistente de ángulos muertos

Figura 1

Uno de los primeros en implementarlo fue Volvo, aunque le han seguido otros como Mercedes-Benz o generalistas como Ford, Opel, Citroën, Volkswagen y tantos otros cada vez más, aunque no siempre sea un equipamiento de serie.

Los sistemas de detección de ángulo muerto consisten básicamente en un elemento electrónico que "ve" lo que el espejo no es capaz de "ver", y una unidad de procesamiento que actúa en consecuencia emitiendo una alerta al conductor. Dependiendo de la marca se han usado distintos sensores para lograr el objetivo. Se ha realizado la detección mediante cámaras, radares, o sensores de ultrasonido.

Una vez detectado el vehículo se trata de trasmitirle al conductor que hay algo que no ve. Algunas marcas lo hacen mediante pantallas de video dentro del coche, y otras mediante una luz o un pitido.

2.2.9 Detección y clasificación de la información vial

Se puede dividir en dos partes, detección de la señalización vertical y horizontal.

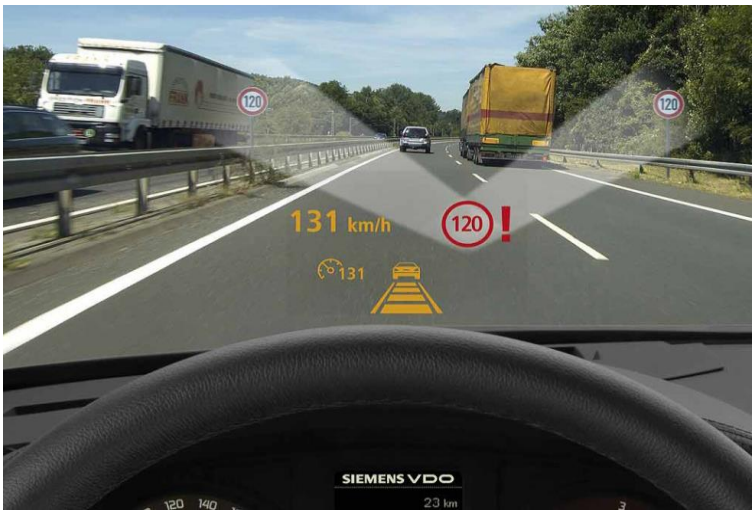


Figura 14

La señalización vertical se refiere a las señales de tráfico. La detección se realiza mediante visión por computador y la clasificación se lleva a cabo por correlación o por redes neuronales.

Este sistema tiene ventajas e inconvenientes. La mayor ventaja es que las señales de tráfico son iguales para toda Europa ya que siguen la normativa europea. Las desventajas son

las típicas de la visión por computador, se necesita una buena visibilidad para que la detección funcione.

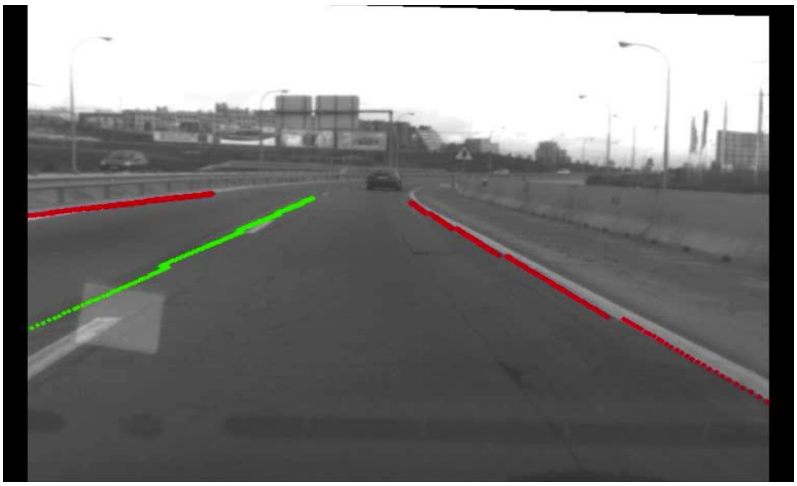


Figura 15

La señalización horizontal se refiere a las pintadas en la carretera. En este caso el objetivo es el análisis de la calzada. Mediante visión por computador (cámara estéreo) se localizarán las líneas de la carretera y se clasificarán según su naturaleza como se puede ver en la foto (verde para discontinua y roja para continua).

El objetivo último de este sistema es un que ayude al conductor a guiar su coche y garantizar que no se salga del carril de manera involuntaria.

2.2.10 Detector de fatiga en el conductor

Figura 16

Este sistema es capaz de reconocer si el conductor está a punto de dormirse al volante, a través de sensores, y de advertirle antes de que ocurra un accidente. El sistema realiza un seguimiento de varios aspectos de la cara, incluido el grado de apertura de los ojos. Además analiza los patrones de conducción y las reacciones del conductor y los combinan con datos sobre la velocidad de circulación, la hora y el comportamiento del intermitente. Si detecta que hay cansancio en el conductor le avisa mediante el símbolo de una taza de café en el salpicadero para que se tome un descanso.

2.2.11 Detección de peatones

Figura 17

Los peatones son los elementos más vulnerables de la circulación. Esta tecnología es capaz de detectar la presencia de un peatón delante del vehículo y si el conductor no responde a tiempo el vehículo avisa y activa automáticamente los frenos. A través de un radar en la parrilla del coche, una cámara al lado del espejo retrovisor interior y una unidad de control central el sistema detecta cualquier peatón situado delante del coche al tiempo que calcula la distancia entre ambos. También es capaz de detectar peatones que están a punto de alcanzar la calzada. El sistema de detección de peatones con frenado automático puede evitar una colisión con un peatón a velocidades de hasta 35 km/h. A más velocidad la prioridad es reducir la velocidad del coche antes del impacto. Para suavizar las consecuencias del impacto Volvo fue el primer fabricante del mundo en incorporar un airbag para peatones.

La detección se puede llevar a cabo con cámaras estéreo, cámaras infrarrojas, escáneres láser, radares, o incluso con varios de estos sensores lo que aumentaría el éxito de la detección.

2.2.12 Detección de vehículos

Figura 18

Al igual que la detección de peatones, se trata de localizar si en nuestra trayectoria se encuentra o se va a encontrar un vehículo u obstáculo con el que podríamos colisionar.

La detección se puede llevar a cabo por múltiples sensores o por una mezcla de ellos, al igual que en la detección de peatones, solo que nuestro modelo a encontrar, en lugar de ser una persona, será un vehículo o cualquier otra cosa que nos interese.

Gracias a la detección de vehículos se han podido desarrollar algunos sistemas interesantes como lo son el "Stop & Go", detiene el coche cuando el vehículo anterior se detiene y arranca cuando este se mueve, muy útil para los atascos, y el "Control de crucero inteligente".

Figura 19

2.3 Coche de Google

El automóvil sin conductor de Google (en inglés Google driverless car) es un proyecto de Google consistente en el desarrollo de la tecnología necesaria para crear coches sin conductor, que circulen de forma autónoma. Actualmente el líder del proyecto es el ingeniero alemán de Google Sebastian Thrun, director del Stanford Artificial Intelligence Laboratory y coinventor de Google Street View. El equipo de Thrun en Stanford creó el vehículo robótico Stanley, que fue el ganador del DARPA Grand Challenge en 2005, otorgado por el Departamento de Defensa de los Estados Unidos y dotado con un premio de 2 millones de dólares. El equipo encargado del proyecto estaba formado por 15 ingenieros de Google, entre los que se encontraban Chris Urmson, Mike Montemerlo, and Anthony Levandowski, quienes habían trabajado en el DARPA Grand and Urban Challenges.

Este coche es capaz de conducir autónomamente por ciudad y por carretera, detectando otros vehículos, señales de tráfico, peatones, etc.

Figura 20

El Stanley está construido sobre el Volkswagen Touareg estándar Europeo. Se eligió este modelo por su sistema de control "drive by wire" el cual consiste en sustituir los actuadores mecánicos del vehículo por actuadores electro-mecánicos o simplemente eléctricos.

Para poder navegar, el Stanley monta cinco sensores LIDAR en el techo. El LIDAR es un sensor que es capaz de recrear el entorno en 3-D al igual que el Radar, pero a diferencia de éste se consigue mediante luz reflejada y no ondas sonoras. Tienen un alcance de hasta 60 metros y 360 º, lo cual es realmente importante.

Estos sensores complementan a la navegación por GPS que es fundamental. También incorpora una serie de sensores como giróscopos y acelerómetros que monitorizan la orientación del vehículo y sirven para complementar al GPS.

Además, una cámara proporciona datos adicionales de navegación hasta 8 metros de distancia por medio de visión por computador la cual complementa a los anteriores sensores mencionados.

Por último, el Stanley incorpora unos sensores en las ruedas los cuales entran en juego en caso de que se pierda la señal de navegación por GPS. Estos sensores actúan como calculadores de distancia recorrida desde el momento en que se pierde la señal para así poder realizar el seguimiento del vehículo aún en caso de una pérdida de señal.

Todos los datos de los sensores son procesados por un computador equipado con seis Intel Pentium M a 1.6 GHz de bajo consumo, trabajando en diferentes versiones de Sistema Operativo Linux.

A día de hoy google ha dado un paso hacia delante y ha decidido que es hora de dejar de montar sus tecnologías en otros modelos y crear su propio prototipo. Se trata de un pequeño biplaza de formas redondeadas y no tiene volante, acelerador ni frenos.

Figura 21

A diferencia de los Prius y Lexus con los que google ha estado trabajando, este prototipo no necesita más interacción humana que la de definir un destino sobre Google Maps.

El vehículo cuenta con sensores que analizan cientos de objetos simultáneamente en 360 º y a 180 metros de distancia.

Google fabricará un centenar de estos prototipos a los que dotará de controles convencionales para dar mayor seguridad a sus pasajeros. Las pruebas empezarán en California y puede que se extiendan durante años. La estrategia de Google no es poner este coche a la venta ni en un futuro cercano ni lejano, si no perfeccionar el prototipo para poder vender la tecnología a los fabricantes de automóviles.

2.4 Sistema operativo Android

Android es un sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, y también para relojes inteligentes, televisores y automóviles, inicialmente desarrollado por Android, Inc. Google respaldó económicamente y más tarde compró esta empresa en 2005. Android fue presentado en 2007 junto la fundación del Open Handset Alliance, un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.

Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto a escala mundial alcanzó una cuota de mercado del 50,9% durante el cuarto trimestre de 2011, más del doble que el segundo sistema operativo (iOS de Apple, Inc.).

En las siguientes gráficas podemos ver cómo ha evolucionado la cuota de mercado para los años 2012 y 2013.

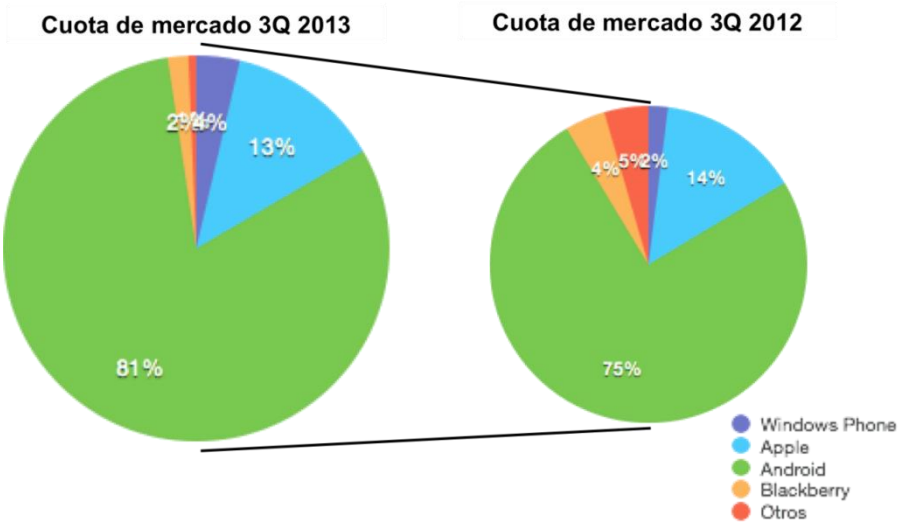


Figura 22

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik concompilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager).

El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

Como podemos ver el sistema operativo Android tiene apenas 7 años de vida. Esto no significa que nos encontremos ante un sistema operativo incompleto, es más, Android es un sistema operativo maduro utilizado por millones de personas en el mundo. Pero si lo comparamos con otros sistemas operativos normalmente usados en la visión por computador como lo son Windows o, el más importante, Linux, nos damos cuenta que la primera versión de Windows data del año 1985 y la de Linux de 1991.

Esta diferencia de vida entre los S.O. tradicionales y Android significa mucho tiempo de evolución y mejoras y nos da una idea de que aunque Android sea a día de hoy un S.O. de pleno derecho, aún le quedan muchas cosas por incluir y mejorar.

Haciendo referencia al tema concreto del proyecto, la visión por computador, Android es un S.O. el cual no está orientado hacia este tipo de tareas, o mejor dicho, está empezando con ello.

Gran parte de la culpa de dicho inicio la tienen las librerías OpenCV recientemente adaptadas para Android. La primera versión lanzada para Android data de Julio del 2011, lo cual es una vida relativamente corta con respecto a las primeras versiones para Windows, Linux, y Mac que datan del año 2006.

Además de una vida relativamente corta, hay que considerar las limitaciones de hardware de los dispositivos que usan Android. Debido a estas limitaciones, las funcionalidades de la versión para Android de OpenCV son también limitadas. El procesamiento de imágenes es, de las tareas que puede llevar a cabo un dispositivo, de las que más cantidad de procesamiento necesita. Por ello, no tendría sentido añadir funcionalidades las cuales no pueden ser llevadas a cabo por los actuales dispositivos móviles que son los que usan Android.

Las librerías OpenCV están escritas en C++ mientras que Android se programa en Java. Dicha adaptación consiste en una serie de librerías para traducir las librerías originales escritas en C++ a Java, las cuales ya pueden ser usadas para programar.

Dicho todo esto, queda bastante claro que los resultados que se conseguirán llevando a cabo un procesamiento de imágenes con Android no será igual de satisfactorio que si se llevara a cabo en un PC convencional el cual dispone de hardware mucho más potente y un S.O. especializado en dichas tareas, como puede serlo Linux.

2.5 Aplicaciones para la ayuda a la conducción

A pesar de los inconvenientes mencionados anteriormente sobre el procesamiento de imágenes en Android, se han llevado a cabo distintos proyectos los cuales tienen como objetivo que nuestro dispositivo móvil nos ayude en la conducción.

Dichas aplicaciones intentan imitar lo que ya está mucho más desarrollado para otras plataformas. El inconveniente de estas aplicaciones es que siempre funcionaran peor que si incorporamos un PC en nuestro vehículo, pero tienen la gran ventaja que el dispositivo móvil es algo pequeño que siempre llevamos encima. Además los smartphones han tenido un gran boom en los últimos años, por lo que prácticamente todo el mundo tiene uno.

A continuación se indicarán algunas de estas aplicaciones ya existentes y se explicará cómo se pretende que ayuden a la conducción.

2.5.1 Waze.

Recién adquirida por Google, es la aplicación de conducción más famosa y con más éxito de todos los sistemas operativos para móviles. Tiene 40 millones de usuarios e informa del estado del tráfico desde Israel a España, pasando por Inglaterra o Estados Unidos. Es gratuita.

Se basa en el GPS del teléfono para compartir nuestra ubicación en una red social de conductores, donde el sistema informa de retenciones, tráfico lento, etc. según detecte la velocidad del vehículo.

Con el móvil colocado a modo de navegador GPS, podemos ver en la pantalla un mapa de nuestra ruta con los atascos que nos encontraremos en el camino según vayan ocurriendo. Nuestras propias retenciones las podemos compartir con nuestros contactos de Facebook además de en Waze.

2.5.2 iOnRoad Augmented Driving.

Figura 23

Este programa utiliza las propiedades del teléfono que permiten la realidad aumentada, así como los diversos sensores y la cámara, para avisarnos de posibles colisiones con otros vehículos y otros peligros.

El móvil se coloca con un soporte en el salpicadero, de modo que permita enfocar la cámara al frente a través de la luna delantera. Así transmite la conducción en tiempo real y hace que aparezcan en la pantalla diferentes datos, como la distancia de seguridad recomendada y la que realmente guardamos, o si excedemos la velocidad o nos salimos del carril.

2.5.3 Drivea.

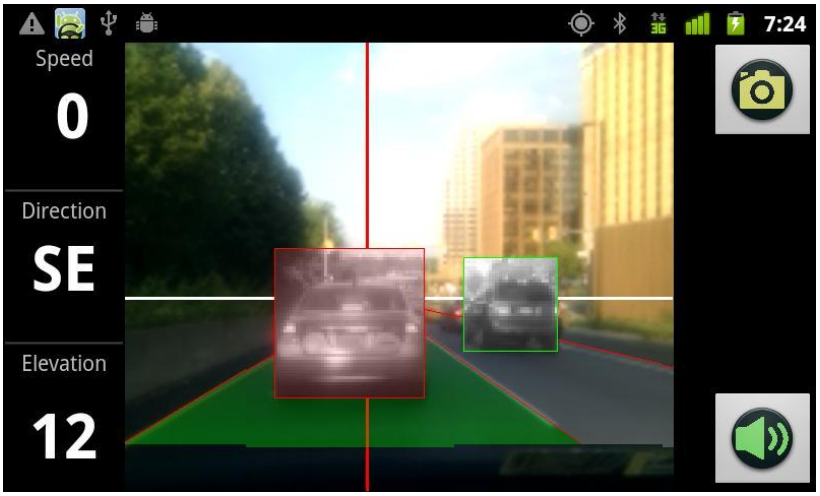


Figura 24

Similar a la anterior aplicación, este software está más centrado en calibrar los riesgos que se puedan producir y en advertir de si hay peligro de colisión o nos salimos de carril o si sobrepasamos la velocidad límite. También usa los sensores del teléfono para calcular los peligros. Se sitúa con un soporte sobre el salpicadero y frente a la luna.

2.5.4 Car Black Box ALPHA

Esta app actúa de caja negra del coche, al emplear todos los sensores del móvil para registrar posición, ruta, velocidad en cada momento, etc. Todos estos datos los guarda para que, en caso de accidente, puedan saberse con veracidad los datos del usuario.

Es útil para un juicio donde se nos acuse de ir a una velocidad inadecuada, para nuestra compañía de seguros o para recurrir multas. Además, en caso de accidente en el que quedemos inconscientes, detecta el siniestro y envía un aviso a nuestros contactos de que podemos estar heridos.

2.5.5 Car Home Ultra

Figura 25

Parecido a iOnRoad Augmented Driving, ofrece datos del coche y nuestra conducción en base a los sensores del teléfono y a la realidad aumentada. Sin embargo, Car Home Ultra se centra más en datos técnicos que en los riesgos, aunque también avisa por si nos pasamos de la velocidad recomendada. Estudia nuestro consumo energético, la meteorología en el trayecto, la altitud o la distancia con los posibles destinos. Es, en definitiva, un navegador GPS enriquecido. Además, puede contestar llamadas en las que se indica que estamos en plena conducción.

2.5.6 Safe Driver

Este programa es similar a “I'm Driving” aunque algo más completo. Detecta nuestra actividad por el GPS y los sensores de movimiento y si entiende que estamos conduciendo, bloquea el móvil y se encarga de recibir y gestionar desde llamadas a SMS o notificaciones de WhatsApp o Facebook Messenger, etc. Responde a todos con un mensaje corto que indica nuestro estado. Y cuando nos detenemos por un tiempo prolongado, nos informa de las comunicaciones que hemos recibido.

3. Descripción general

3.1 Propósito

El objetivo del proyecto es el cálculo de distancias usando un dispositivo móvil. Para ello este proyecto ha sido trabajado sobre otro ya existente el cual consistía en detectar peatones usando la cámara de video de un Smartphone. Este proyecto lleva eso más allá calculando la distancia existente desde el dispositivo móvil hasta ese objeto (peatón) detectado. Esto abre la puerta a otros posibles proyectos que tengan el objetivo de mejorar y complementar este de múltiples formas.

Este proyecto está enfocado, tanto para las explicaciones como las pruebas realizadas, en la detección de peatones, pero existe un gran abanico de posibilidades en los que implementar este algoritmo de cálculo de distancias.

En un ámbito automovilístico este algoritmo se podría implementar, además de con peatones, con otros vehículos, con señales de tráfico, con obstáculos en la calzada de distinta naturaleza. Este algoritmo se podría considerar un primer paso que deja abierta la posibilidad de seguir con un proyecto de mayor envergadura. El objetivo final sería una aplicación capaz de localizar y seguir en el espacio un objeto, ya sea peatón, vehículo, o cualquier otra cosa que deseemos. Con esto podríamos llevar a cabo gran cantidad de tareas como podría ser un aviso por sonido si un objeto se nos podría cruzar en nuestra trayectoria, o una adaptación de velocidad para poder seguir a otro vehículo a una distancia prudente, detectar si hay un vehículo en el carril contiguo que pueda invadir nuestro carril, etc.

Fuera del ámbito automovilístico también se podría usar este algoritmo de cálculo de distancia en móviles para otras tareas de distinto tipo como por ejemplo un metro genérico de gran precisión (ya que nosotros elegiríamos manualmente el punto donde medir) o un detector de cualquier tipo de objeto al que calculáramos la distancia.

3.2 Hardware

En este apartado se describirán todos los componentes de hardware que necesita el algoritmo para funcionar correctamente. Estos serán el Smartphone y sus sensores.

En la ayuda a la conducción en Smartphone tenemos grandes limitaciones a diferencia de otros dispositivos de ayuda a la conducción. En otros dispositivos se usan sensores de gran precisión como el radar, lidar, laser, cámaras estereo, cámaras infrarrojas, y computadores con gran capacidad de cálculo. En la ayuda a la conducción en Smartphone solo disponemos de los limitados sensores integrados en el dispositivo móvil. Afortunadamente estos sensores son

suficientes para realizar una detección y un cálculo aproximado de la distancia, con sus limitaciones claro está.

En este proyecto los sensores utilizados son la cámara de fotos/video integrada y el acelerómetro.

3.2.1 Cámara

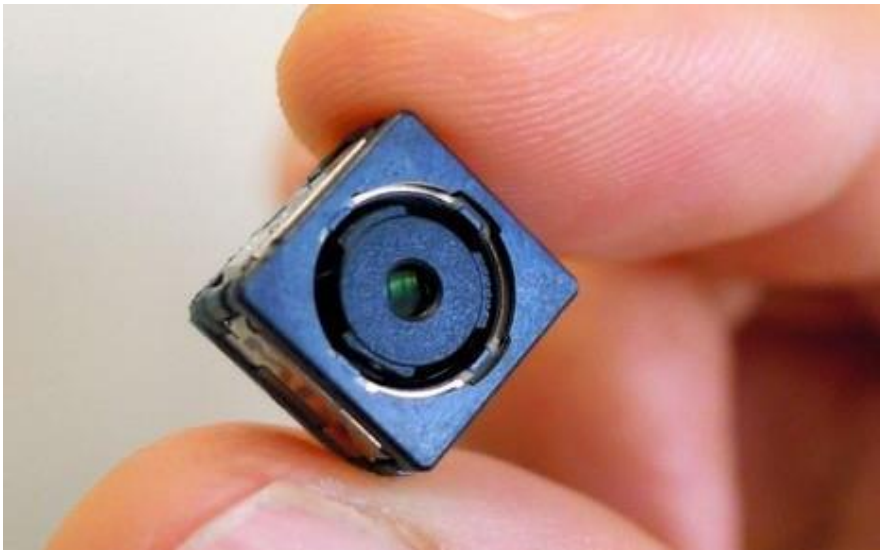


Figura 26

La cámara es necesaria para capturar el entorno y poder analizarlo. En las imágenes captadas se lleva a cabo un procesamiento de la imagen para realizar la detección del obstáculo y en ella también se analizan los parámetros necesarios para el cálculo de la distancia.

De la cámara, además de la propia captura de la imagen, se necesitan sus parámetros intrínsecos, los cuales son la distancia focal, el tamaño del sensor, y el punto central de la imagen. Estos parámetros intrínsecos se consiguen con las herramientas de programación que nos proporciona Android, las cuales son muy limitadas en este ámbito. El proceso para averiguar dichos parámetros quedará explicado en el apartado “Algoritmos”.

3.2.2 Acelerómetro

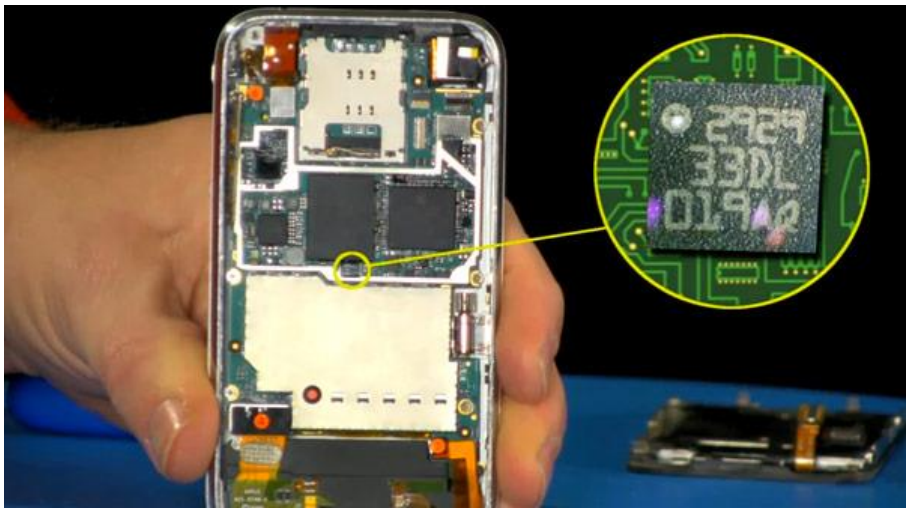


Figura 27

El acelerómetro será necesario para el cálculo de distancias. El acelerómetro indica que aceleración está sufriendo el dispositivo móvil, descompuesta esta en cada eje de coordenadas.

Para el cálculo de la distancia no interesa la aceleración, lo que interesa es la orientación que tiene el dispositivo en el momento justo de la captura de la imagen. Esto se consigue mediante la aceleración de la gravedad, la cual habrá que aislar de la aceleración puntual que pueda sufrir el dispositivo en un momento dado. Este proceso quedará explicado en el apartado “Algoritmos”.

Sabiendo la dirección a la que apunta la gravedad descompuesta en cada eje, seremos capaces de calcular los distintos ángulos de Euler necesarios para realizar las correcciones en el cálculo de las distancias. Estas correcciones serán necesarias ya que una pequeña inclinación de la cámara se traduce en una gran diferencia en la imagen por lo que habrá que actuar en consecuencia.

El ángulo de Euler que nos interesa para el proyecto es el llamado “pitch” ya que por distintos motivos los otros ángulos no afectan. El “yawn” no interesa ya que no afecta a la distancia en ningún sentido y el “roll” no se ha implementado ya que la detección deja de funcionar si aplicamos este tipo de rotación al dispositivo.

3.3 Software

Este proyecto está realizado mediante dos software de gran popularidad y aceptación entre los programadores que se dedican a la visión por computador. Por un lado la programación se ha llevado a cabo en el entorno de desarrollo Eclipse y el procesamiento de imágenes se hace mediante OpenCV.

3.3.1 Eclipse

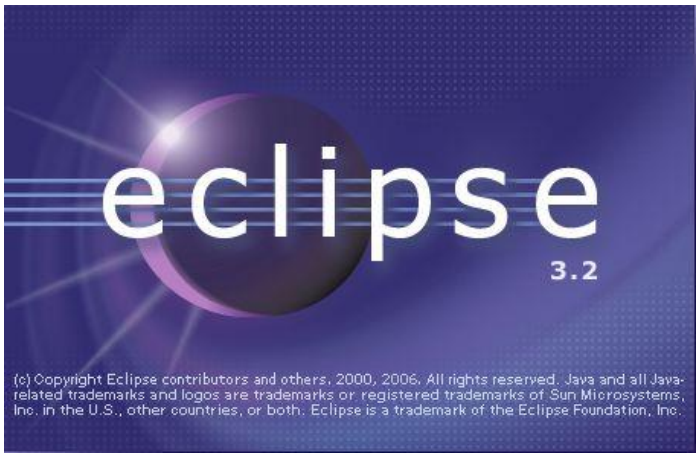


Figura 28

Este proyecto se ha programado en Eclipse con el plugin para programación en android ADT (Android development tool). De este software decir que es de código libre y se ha elegido por delante de otros específicos para programación en Android por su gran versatilidad para el uso de herramientas externas a él, como por ejemplo lo son las librerías OpenCV, las cuales son la base de este proyecto. También lo he elegido ya que tenía un interés personal en aprender a usarlo ya que es un entorno de desarrollo de gran versatilidad útil para programar en gran cantidad de lenguajes.

3.3.2 OpenCV

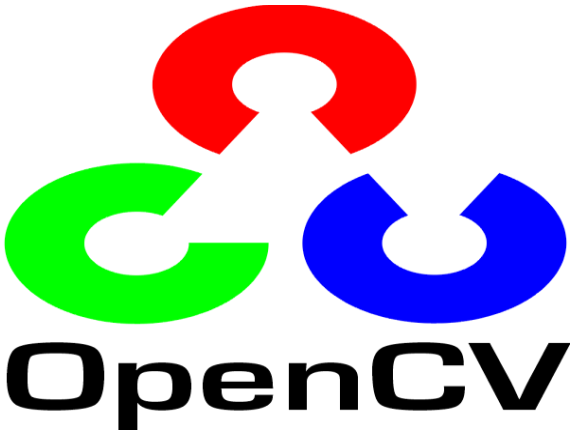


Figura 29

Las OpenCV son unas librerías para el procesamiento de imágenes desarrolladas por google de código abierto. En este proyecto se ha usado la versión creada recientemente para Android basadas en lenguaje Java. Esta versión es mucho menos completa que la versión basada en el lenguaje C/C++ ya que son relativamente jóvenes con respecto a estas otras y aún no se han implementado todas las características existentes en estas. La razón de esto puede ser la menor capacidad de cálculo de un Smartphone con respecto a un PC convencional por lo que lo han intentado adaptar para tener las funciones básicas de estas librerías pero sin sobrepasar el límite aceptado por el hardware.

4. Algoritmo

El algoritmo es un conjunto de ecuaciones mediante las cuales podemos calcular la distancia desde nuestra posición hasta el objeto detectado. Para este cálculo de la distancia necesitamos una serie de parámetros. Estos se pueden diferenciar en parámetros intrínsecos a la cámara y extrínsecos a esta.

Los parámetros intrínsecos de la cámara son aquellos inherentes a ella y que son distintos dependiendo el dispositivo móvil que estemos usando, ya que cada uno monta en él cámaras distintas.

Los parámetros extrínsecos son otra serie de parámetros necesarios para nuestro algoritmo externos a la cámara. Estos pueden ser, por ejemplo, parámetros que nos indican la orientación del dispositivo o puntos de la imagen tomada.

Para la explicación del algoritmo se procederá a explicar en un primer momento como se hallan estos parámetros necesarios para el algoritmo y luego se procederá a explicar el algoritmo en sí mismo.

4.1 Acceso a sensores

Tanto para los parámetros intrínsecos como para alguno extrínseco es necesario acceder a los sensores del dispositivo móvil para poder adquirir los valores numéricos de dichos parámetros. Accederemos tanto a la cámara como al acelerómetro mediante programación en Android. Usaremos las librerías de funciones que nos proporciona Android para conseguir dichos valores. Se explicará en detalle el acceso a cada parámetro en su apartado específico.

4.2 Parámetros intrínsecos

Los parámetros intrínsecos necesarios para nuestro algoritmo son la distancia focal, el tamaño del sensor, y el punto central de enfoque de la imagen.

El algoritmo se basa en el modelo pinhole, por ello tanto las definiciones como las ilustraciones se corresponderán con este modelo.

4.2.1 Distancia focal

Para el modelo pinhole, la distancia focal se define como la distancia entre la película y agujero por donde se dejan pasar los rayos. En nuestro caso, la distancia focal será la distancia entre el centro óptico de la lente y el punto focal (donde se concentran todos los rayos).

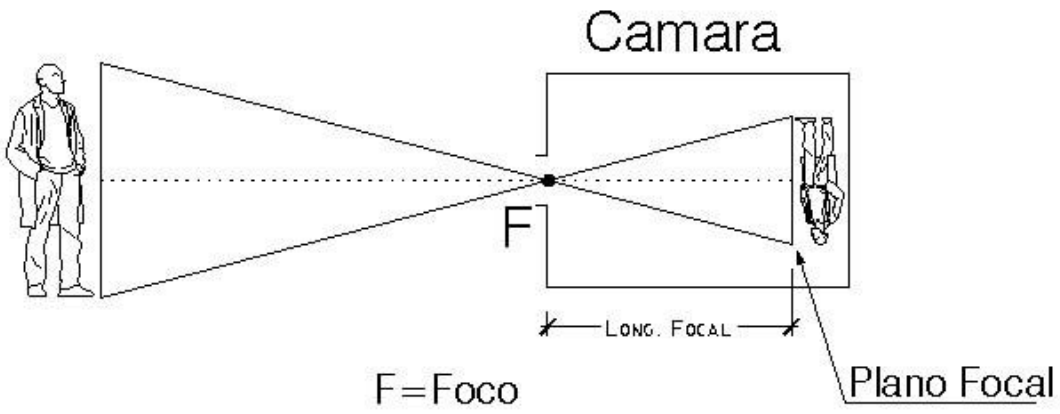


Figura 30

Este valor lo tendremos que conseguir para cada dispositivo ya que cada cámara es distinta. Se realizará mediante programación antes de que empiece a funcionar el procesamiento de imágenes en sí mismo. Para ello tendremos que acceder a la clase "Camera" y usar una de las funciones que nos proporciona Android "getFocalLengh()". Esta función nos devolverá el valor de la distancia focal en milímetros.

4.2.2 Tamaño del sensor

Es el tamaño del sensor de la cámara en el cual se encuentran en su superficie los sensores sensibles a la luz. Cada uno de estos sensores corresponde a un píxel en la imagen por lo que a mayor número de ellos más megapíxeles tendrá la imagen capturada y por tanto mayor definición. Este tamaño será distinto dependiendo si queremos el tamaño en vertical o en horizontal ya que el sensor no es cuadrado si no rectangular, normalmente en un formato de 16:9 o 4:3.

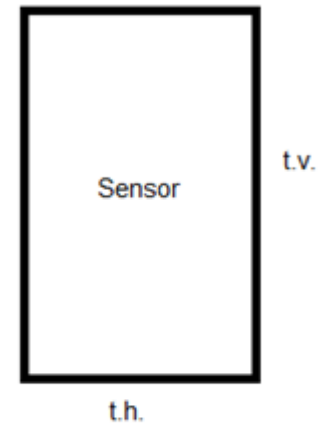


Figura 31

A diferencia de la distancia focal, Android no nos proporciona una función que nos devuelva el valor del tamaño del sensor por lo que tendremos que hallarlo de forma indirecta. Mediante un par de funciones que nos devuelven el ángulo vertical y horizontal de visión de la cámara y sabiendo la distancia focal se puede calcular el tamaño del sensor mediante un proceso trigonométrico simple.

Figura 32

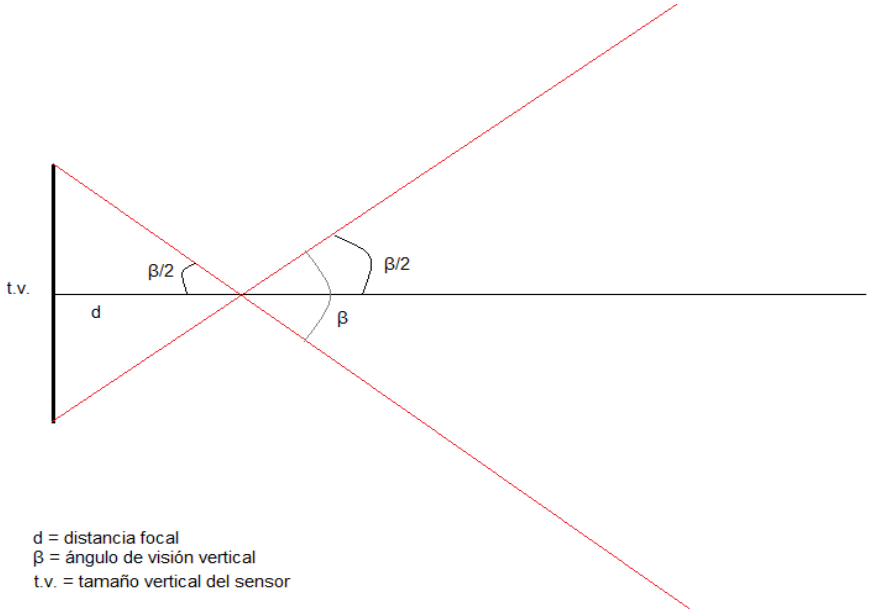


Figura 33

$$\tan\frac{\beta}{2} = \frac{t.v/2}{d}$$

$$t.v = 2d \times \tan\frac{\beta}{2}$$

El ángulo de visión horizontal sería equivalente pero situado en el plano horizontal. Necesitaremos ambos tamaños para los cálculos posteriores.

4.2.3 Punto central de enfoque de la cámara

Se trata del pixel concreto el cual no sufre refracción de la lente ya que se encuentra en el centro exacto del sensor. En las cámaras especializadas para visión artificial es un parámetro que viene siempre indicado y suele coincidir con el punto central de la imagen, o con uno muy cercano a este.

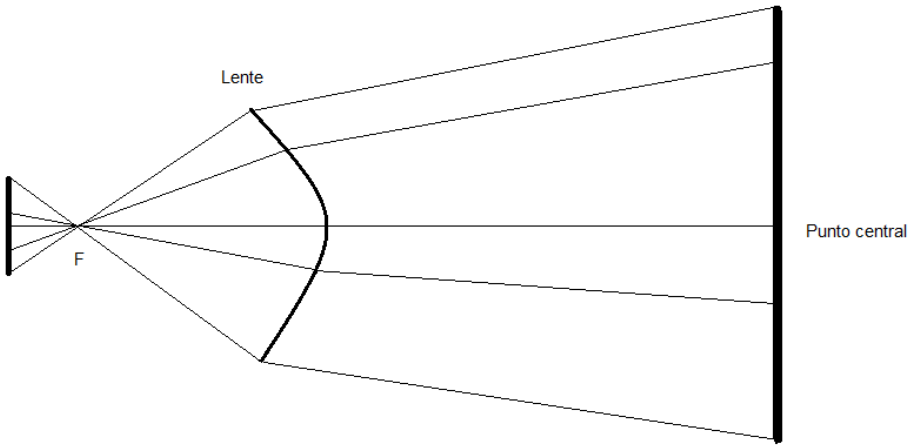


Figura 34

Como podemos observar en la imagen, todos los rayos de luz, excepto el central, sufren una desviación debida a la lente.

Como Android no nos proporciona ningún tipo de información al respecto se ha tomado este pixel como el central de la imagen, pudiendo o no derivar en un pequeño error en los cálculos de la distancia dependiendo de la cámara.

4.3 Parámetros extrínsecos

4.3.1 Cálculo del Pitch

El “Pich” es uno de los tres ángulos que define Euler para determinar la orientación de un objeto en el espacio con respecto a un eje de coordenadas fijo de referencia.

Para este proyecto el eje fijo de coordenadas que se ha tomado es el suelo. La única información que podemos obtener sobre este eje fijo es la dirección de su eje Z perpendicular al suelo estando este indicada por el vector gravedad.

Debido a esto no podremos calcular otro de los ángulos de Euler llamado “Yawn”, ya que carecemos de información del plano perpendicular a la gravedad, pero si se podrá calcular tanto el “Roll” como el “Pitch”.

La dirección del vector gravedad se consigue mediante el acelerómetro. El acelerómetro es un sensor que llevan incorporado todos los Smartphone y que indica la aceleración total que está sufriendo el dispositivo en cada instante. Este dato nos lo

proporciona Android dividido en sus tres componentes según el eje de coordenadas inherente al dispositivo móvil. Para acceder a este dato se usa la programación para acceder a los datos del sensor y se le indica al dispositivo un tiempo de muestreo para que nos actualice el dato según nuestras necesidades.

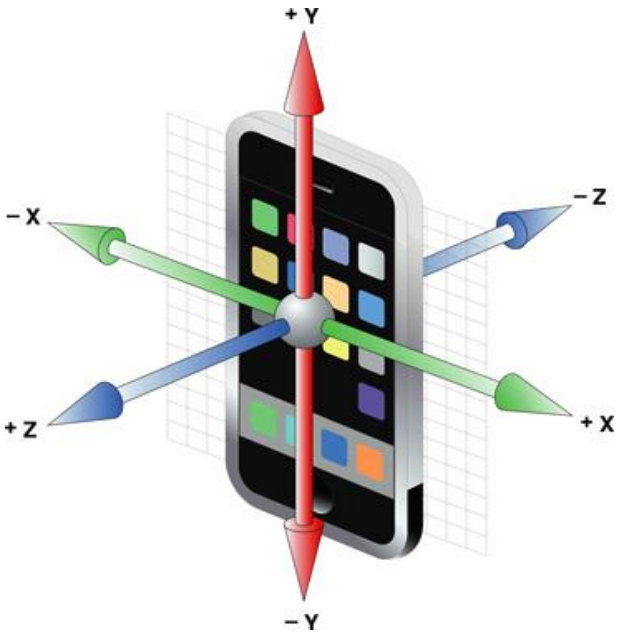


Figura 35. Eje de coordenadas estandarizado de un Smartphone

Para aislar la gravedad del resto de aceleraciones puntuales que pueda estar sufriendo el dispositivo en un momento dado aplicamos un filtro paso bajo, el cual deshecha las variaciones en la aceleración de carácter no permanente y mantiene las de carácter permanente, es decir, la gravedad.

Las cuentas se llevaran a cabo sobre cada componente de la aceleración de la siguiente manera.

$$G = \alpha \times G + (1 - \alpha) \times A.T$$

G es la gravedad que queremos calcular.

α es una constante que indica la restricción del filtro. Su valor oscila entre 0 y 1 y cuanto más cercana a uno más restrictivo es el filtro.

A.T es la aceleración total que nos proporciona el acelerómetro.

Una vez que tenemos la gravedad aislada se ha de calcular los ángulos de rotación. Lo primero será realizar un cambio de coordenadas, ya que el dispositivo se va a usar en posición horizontal y no en vertical.

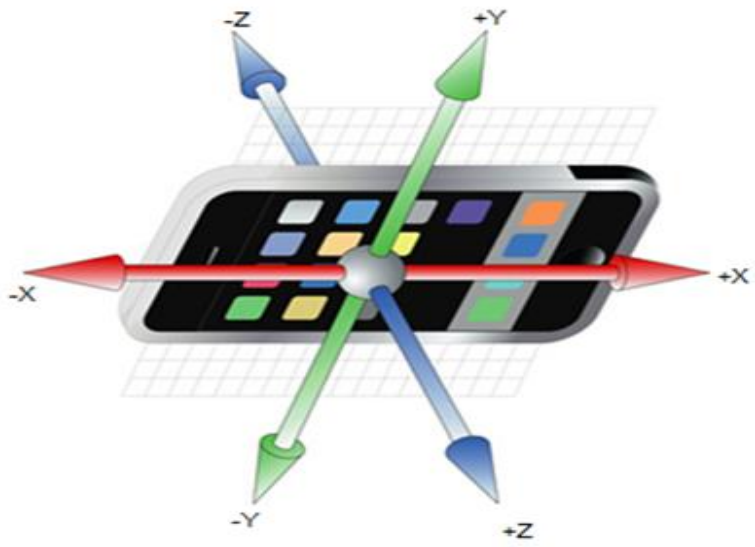


Figura 36. Eje de coordenadas modificado

Con el nuevo eje de coordenadas se define el "Pitch" como la rotación en el eje X, el "Roll" como la rotación en el eje Z, y el "Yaw" como la rotación en el eje Y.

El "Pitch" se calculará de la siguiente manera.

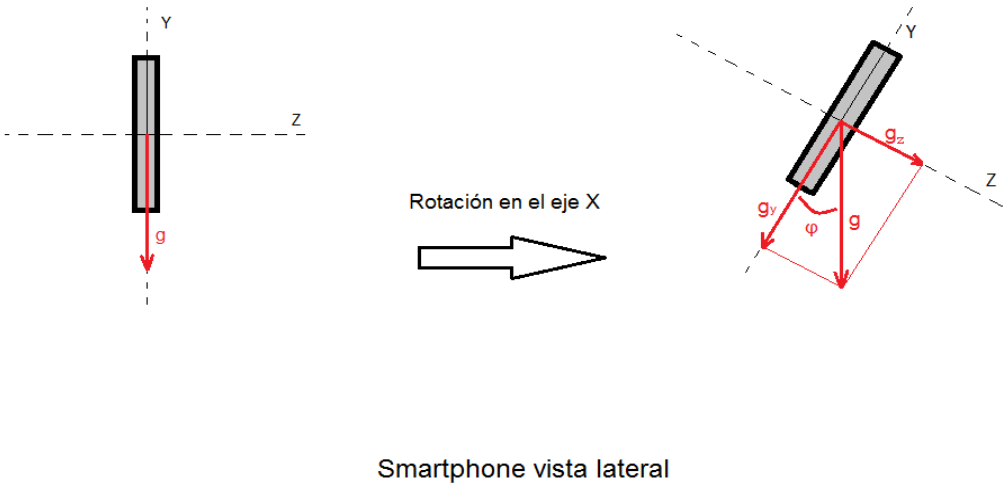


Figura 37

$$\varphi = \cos^{-1}(^{gy}/_{g}) = 90 - \cos^{-1}(^{gz}/_{g})$$

4.3.2 Punto de interés de la imagen

Este es el último parámetro necesario para que el cálculo de la distancia sea posible. Se puede definir como el punto del suelo en el que se sitúa nuestro objeto detectado.

En la aplicación este punto viene dado por la propia detección de peatones, ya que al detectarlo se recuadra el peatón detectado con un rectángulo. Este rectángulo puede ser más o menos ajustado al objeto dependiendo de lo bien que esté hecha la detección. Estos posibles errores de ajuste del recuadro pueden dar lugar a grandes errores en el cálculo de la distancia, por ello es necesario un buen reconocimiento de las formas si queremos que el cálculo funcione bien.

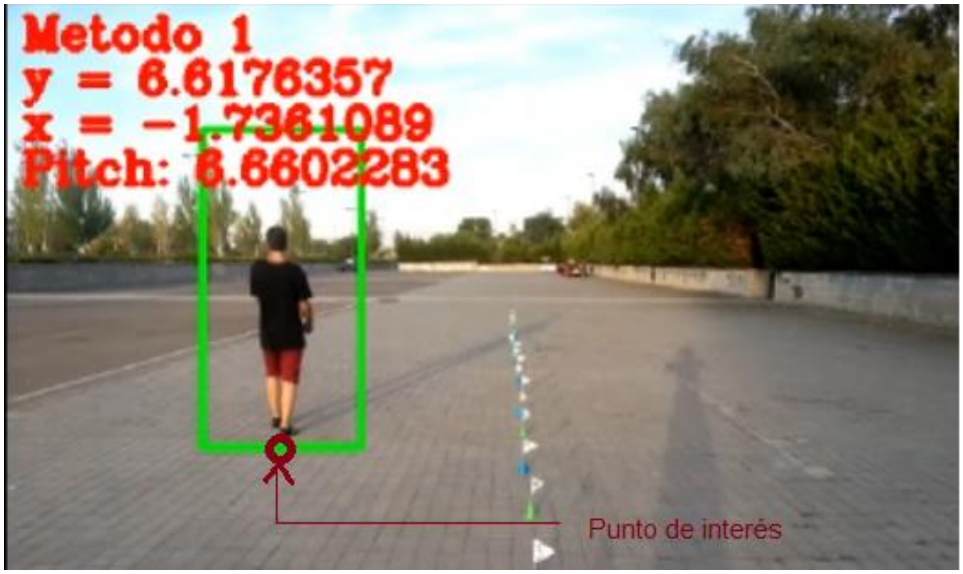


Figura 38

4.4 Cálculo de la distancia

4.4.1 Modelo Pinhole

Figura 39

Se denomina Pinhole a una de las primeras técnicas fotográficas. Esta técnica tiene la peculiaridad de que la cámara no tiene lente, si no que la luz entra hasta la película por un orificio del grosor de una aguja.

La característica principal de las imágenes obtenidas mediante este método es la profundidad de campo infinita. Esto significa que en la fotografía resultante se ve claramente cualquier punto sin necesidad de ningún encuadre especial, como tendría que aplicarse al usar cualquier lente de equipo profesional.

El objetivo del uso del modelo Pinhole es el de conseguir una simplificación de la captura de la imagen para poder usar la trigonometría en el cálculo de distancias y/o tamaños. En la siguiente imagen se puede observar que la imagen real y la impresa en la película guardan una proporcionalidad entre sí dependientes de la distancia focal y de la distancia entre la lente hasta el objeto real.

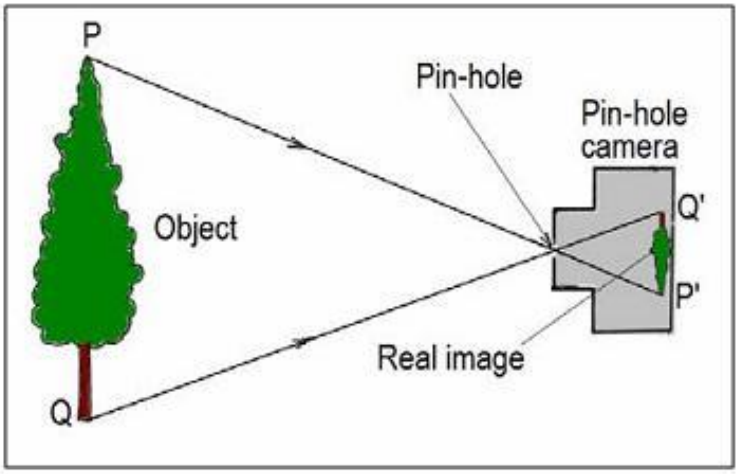


Figura 40

La siguiente imagen muestra la ecuación de proporcionalidad básica del modelo pinhole por triángulos semejantes.

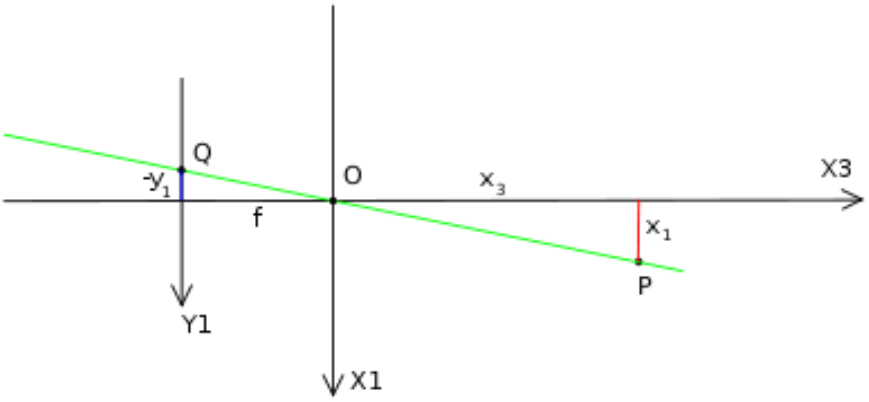


Figura 41

$$\frac{-y1}{f} = \frac{x_1}{x_3}$$

La parte a la izquierda del eje X1 representa el interior de la cámara, siendo f la distancia focal, Y1 el sensor de la cámara, e y1 el tamaño que ocupa nuestra parte de la imagen de interés en el sensor.

La parte a la derecha del eje X1 representa el mundo real, siendo x3 la distancia desde la cámara hasta el objeto de interés que estamos midiendo y x1 el tamaño de dicho objeto.

Analizando la ecuación nos damos cuenta que necesitamos al menos conocer 3 de las incógnitas para calcular una cuarta. La distancia focal es un parámetro ya conocido ya que lo hemos calculado previamente.

Es decir, si queremos calcular la distancia a un objeto tenemos que conocer el tamaño de dicho objeto y el tamaño proporcional de dicho objeto en el sensor y, si queremos calcular el tamaño de dicho objeto, necesitamos conocer la distancia que nos separa de él y el tamaño que ocupa en la imagen.

Se ve claro que ya que el objetivo de este trabajo es calcular la distancia, esa será la incógnita y todos los demás datos tendrán que ser conocidos. A priori, se puede pensar que en el caso que nos atañe no tenemos dichos datos ya que no conocemos el tamaño del peatón ya que este es variable. Esto es totalmente verdad por lo que se usara la siguiente técnica para poder realizar el cálculo de la distancia.

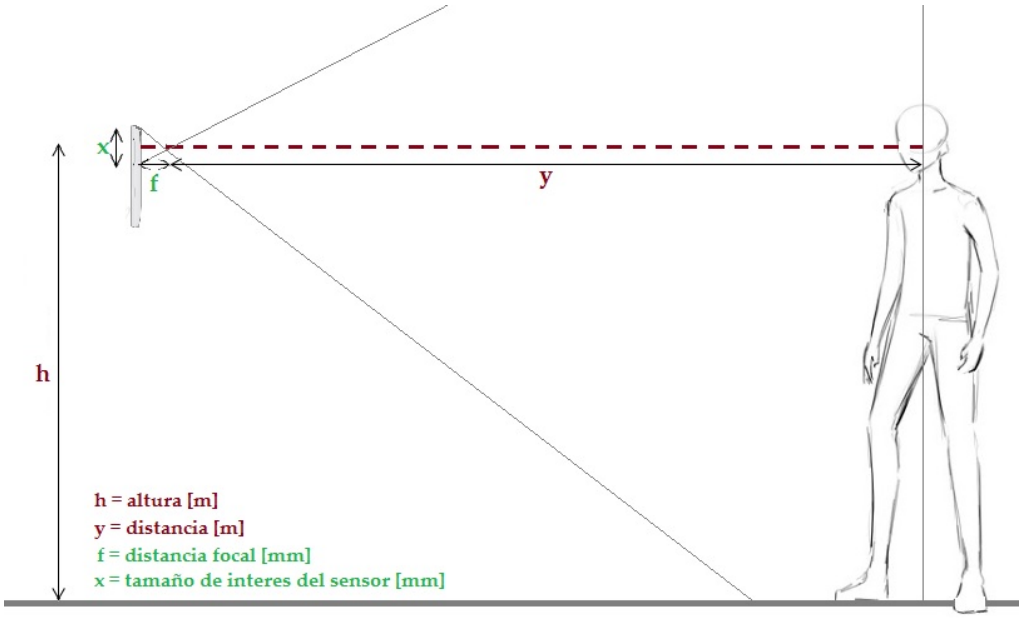


Figura 42

$$\frac{f}{x} = \frac{y}{h}$$

$$y = \frac{f * h}{x}$$

Ecuación principal del modelo pinhole.

A pesar de no conocer el tamaño del peatón, sí que es conocido un dato que va a resultar muy útil. Se trata de la altura (h) a la que estará situado el dispositivo. Como el dispositivo irá fijado en la luna del coche esta altura será una constante conocida.

Suponiendo que el dispositivo está situado a dicha altura y no tiene ningún tipo de rotación, se deduce fácilmente que el rayo de luz que no sufre refracción debido a la lente (explicado anteriormente en la pag. 35) será paralelo al suelo.

De esta manera sabremos que el tamaño desde un punto situado en el suelo hasta el centro de la imagen será siempre la altura a la que esté situado el dispositivo.

Una vez se conoce esto nos encontramos con que se conoce la distancia focal, conocemos el tamaño en la realidad, nuestra h, y también se conoce el tamaño proporcional que ocupa la altura en el sensor ya que viene dado por el punto de interés y el centro de la fotografía.

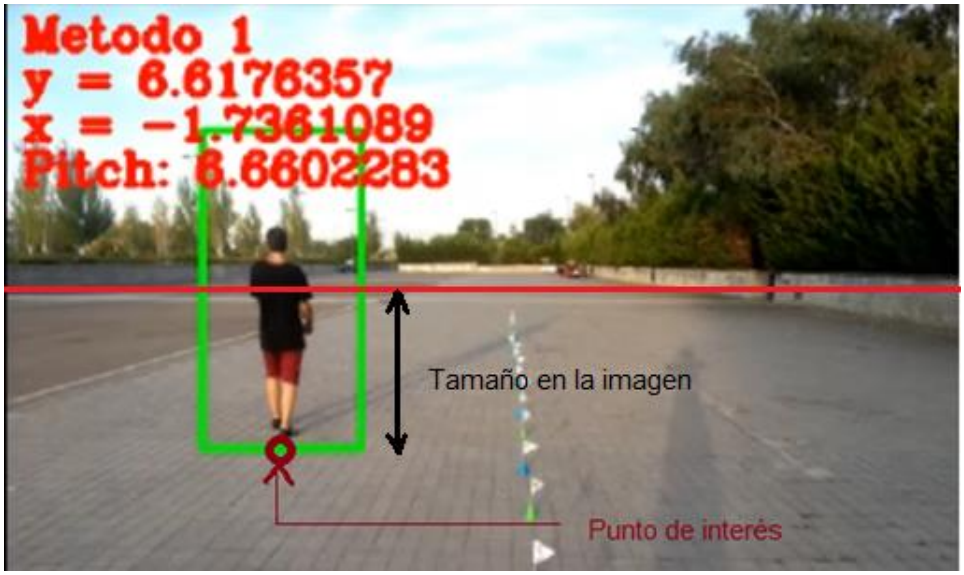


Figura 43

Sabiendo el tamaño en la imagen, en píxeles, es automático convertirlo a tamaño de interés del sensor, en unidades métricas, mediante una proporcionalidad simple.

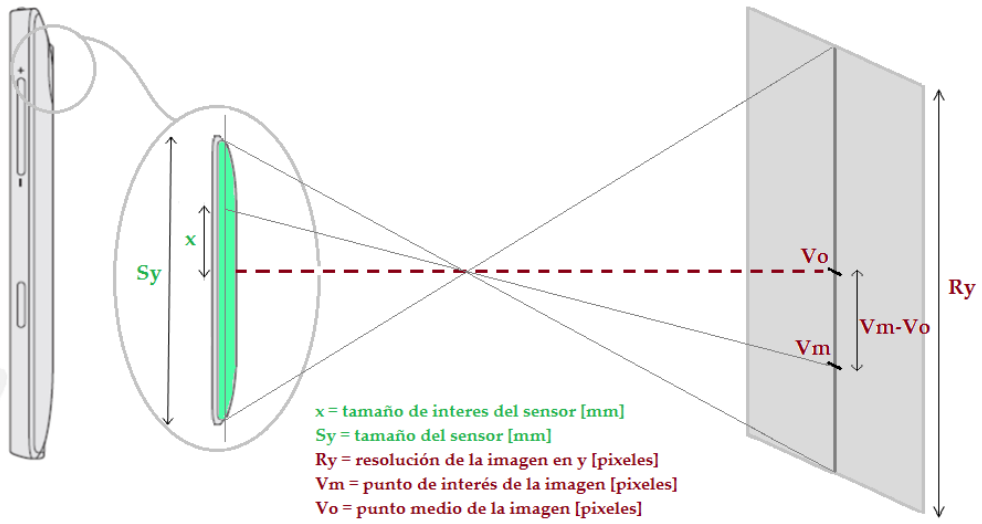


Figura 44

$$\frac{Vm - Vo}{Ry} = \frac{x}{Sy}$$

$$x = \frac{Sy(Vm - Vo)}{Ry}$$

Sustituyendo en la ecuación principal se consigue la siguiente ecuación.

$$y = \frac{f * h * Ry}{Sy(Vm - Vo)}$$

Si se toma la siguiente igualdad.

$$f_y = f\frac{R_y}{S_y}$$

Y se vuelve a sustituir en la ecuación, queda la ecuación final para el cálculo de la distancia.

$$y = \frac{f_y * h}{(Vm - Vo)}$$

La forma matricial para representar al modelo pinole es la siguiente.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ h \end{bmatrix}$$

Para diseñar este método se ha supuesto que el dispositivo no sufre ningún tipo de rotación. Pero en la realidad siempre va a haber una rotación en el dispositivo ya sea pequeña o grande. Con dicha rotación el cálculo de la distancia se ve afectado en gran medida ya que unos pocos píxeles corresponden a una gran distancia en la realidad.

Debido a esto se debe aplicar una corrección para evitar ese error provocado por el "Pitch". Se han probado tres métodos distintos para realizar dicha corrección que a continuación se explicarán uno a uno.

4.4.2 Método 1

Este método se ha extraído del libro "Visión por computador" de Arturo de la Escalera. Consiste en el uso de la trigonometría para calcular la distancia final conociendo el ángulo de rotación (pitch) y la distancia calculada sin tener en cuenta la rotación.

El método se vale de los ángulos para modificar dicha distancia base (sin rotación) de manera que tenga en cuenta el factor rotación y se consiga calcular la distancia real al objeto.

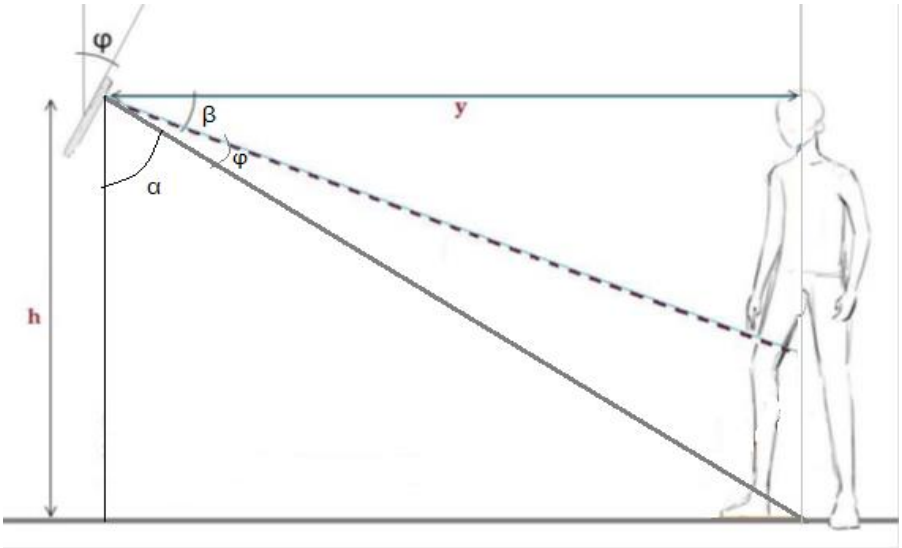


Figura 45

En un primer momento se calcula yo mediante la fórmula del modelo pinole.

$$yo = \frac{f * h * Ry}{Sy(Vm - Vo)}$$

Una vez se tiene yo, se puede calcular el ángulo β mediante la siguiente ecuación.

$$\beta = \tan^{-1}\frac{h}{yo}$$

Como se puede observar en la imagen, el sumatorio de los tres ángulos forman un ángulo recto. Conociendo β y φ (pitch) se podrá calcular el ángulo α de la siguiente manera.

$$\alpha = \frac{\pi}{2} - \beta - \varphi$$

Habiéndose calculado en ángulo α ya se puede calcular la distancia real al objeto detectado de la siguiente manera.

$$y = h * \tan\alpha$$

4.4.3 Método 2

Este método, como el anterior, consiste en modificar el valor de la distancia calculada con la ecuación del modelo pinhole. En este caso la modificación se realizará mediante la aplicación de las matrices de rotación de Euler, las cuales corregirán el error inducido en los cálculos debido a la rotación.

$$R = \begin{bmatrix} \cos\Delta\delta & 0 & \sin\Delta\delta \\ 0 & 1 & 0 \\ -\sin\Delta\delta & 0 & \cos\Delta\delta \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & \cos\Delta\varphi & -\sin\Delta\varphi \\ 0 & \sin\Delta\varphi & \cos\Delta\varphi \end{bmatrix} \begin{bmatrix} \cos\Delta\theta & -\sin\Delta\theta & 0 \\ \sin\Delta\theta & \cos\Delta\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Estas son las matrices de rotación de Euler. La primera indica la rotación en el eje Y, el Yawn, la segunda indica la rotación en el eje X, el Pitch, y la tercera indica la rotación en el eje Z, el Roll.

Las nuevas coordenadas modificadas debido a la rotación se calcularán aplicando las matrices de rotación a las coordenadas calculadas sin tener en cuenta dicha rotación.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

Aplicando la matriz de rotación inherente al "pitch" a la coordenada "y", la cual es la que nos interesa, nos queda la siguiente ecuación para el cálculo de la distancia modificada.

$$y = y_0 * \cos \Delta\varphi - z_0 * \sin \Delta\varphi$$

4.4.4 Método 3

Este método no consiste en realizar una corrección sobre el valor base de la distancia, si no que calcula la distancia teniendo ya en cuenta el ángulo de rotación (pitch).

Para ello se usa la trigonometría modificando el valor de la altura fijado teniendo en cuenta la distancia a la que se sitúa el objeto de interés y el ángulo de rotación.

En la siguiente imagen se puede observar más detalladamente.

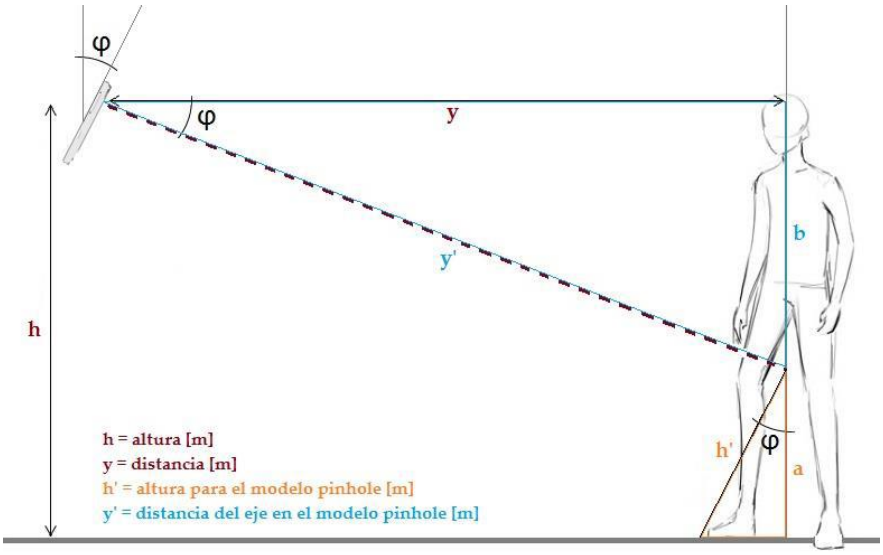


Figura 46

Como se puede observar en la imagen el punto de interés situado en el suelo es invariable, no así lo es el punto central de la imagen ya que el rayo que no sufre desviación ya no es paralelo al suelo, si no que está inclinado un ángulo φ (pitch).

Cuando este rayo incide en el plano en el que se sitúa el punto de interés ya no se encuentra a una distancia h del suelo si no que incide a una nueva altura que llamaremos "a".

El objetivo será el cálculo de esta "a" para introducirla en la fórmula genérica de cálculo de distancia. Habrá que apoyarse en primer lugar en el eje de coordenadas modificado el cual está representado por y' y h'.

Las ecuaciones son las siguientes.

$$\frac{f}{x} = \frac{y'}{h'}$$

$$\frac{f}{x} = \frac{{}^{y}/_{\cos(\varphi)}}{{}^{a}/_{\cos(\varphi)}}$$

$$\frac{f}{x} = \frac{y}{a}$$

Como se ve, al final se anulan los cocientes y conseguimos tener la fórmula en los parámetros que necesitamos.

Por otra parte tenemos que:

$$a = h - b = h - y * \tan(\varphi)$$

Se sustituye en la fórmula anterior y tenemos la ecuación final para el cálculo de distancias con su corrección para la rotación (pitch).

$$y = \frac{fy * h}{(Vm - Vo) + fy * \tan(\varphi)}$$

4.5 Cálculo en el espacio

El objetivo del trabajo no es solo calcular la distancia al objeto, si no llevar a cabo una localización en el espacio completa en el plano del suelo.

Sabiéndose la distancia al objeto se podrían desarrollar ayudas a la conducción relacionadas con las posibles colisiones con peatones situados en la trayectoria del vehículo. Realizando una localización en el espacio se lleva esa idea más allá, dejando abierta la posibilidad a nuevos desarrollos basados en el seguimiento de peatones en el espacio, predicción de movimientos, posibles invasiones de la trayectoria del vehículo, etc.

Para realizar dicha localización serán necesarios dos parámetros, la distancia del peatón al dispositivo en el eje x y la distancia en el eje y.

En la siguiente imagen se explica el concepto.

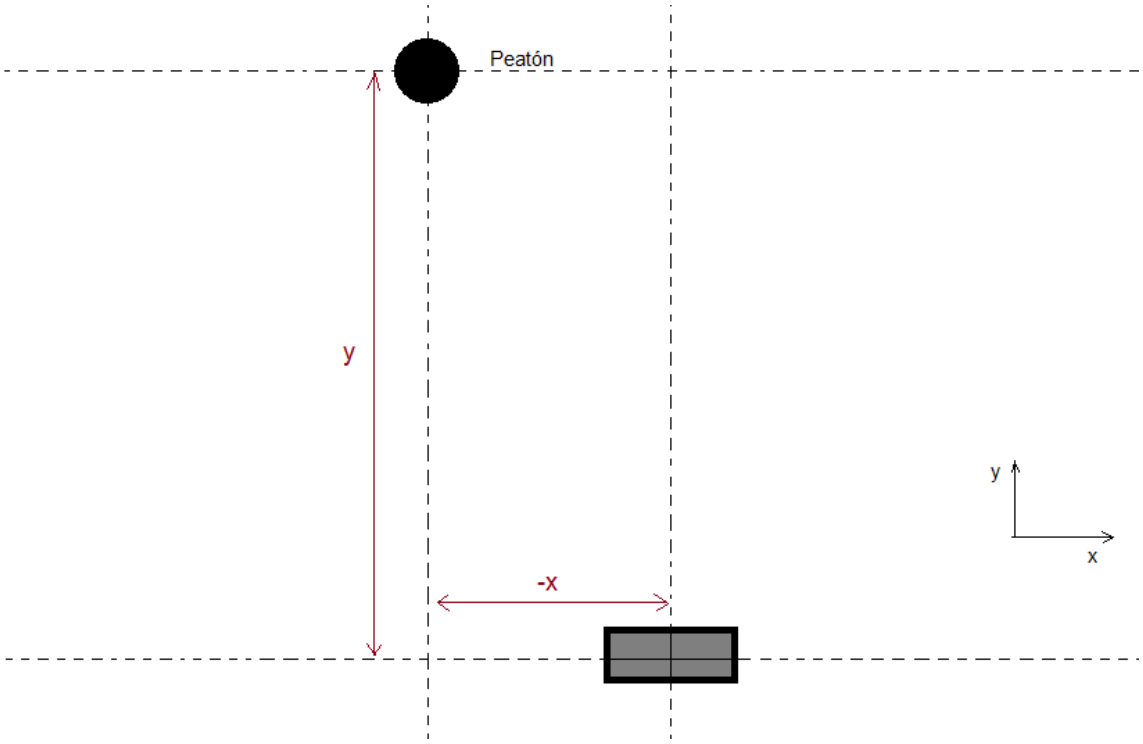


Figura 47

Para poder calcular la componente x se necesita saber la distancia a la que se encuentra el peatón (y) por lo que será el último paso que se realice.

La x se puede calcular despejando la componente en la matriz del modelo pinhole.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ z \\ h \end{bmatrix}$$

$$x = \frac{y(Um - Uo)}{f_x}$$

Siendo Um y Uo las correspondientes Vm y Vo pero en el plano horizontal de la imagen.

Siendo “y” la distancia calculada previamente hasta el objeto.

4.6 Descripción de la biblioteca

La creación de una biblioteca responde a la necesidad de agrupar una serie de funciones, que nos proporcionan las actuales bibliotecas de programación en Android, que puedan ser útiles para realizar un seguimiento de un objeto, así como el cálculo de otros parámetros útiles, como puede ser la orientación del dispositivo. También se ha procedido a la creación de algunas nuevas funciones que no son proporcionadas por Android para poder facilitar el trabajo en un futuro a aquellas personas que quieran trabajar en el tema que se está tratando.

La biblioteca estará formada por un solo atributo y once funciones.

4.6.1 Atributos

- *Float Height*

Se tratará de una constante la cual indicará la altura a la que se encuentra el dispositivo. Será necesaria para llevar a cabo muchos de los cálculos usados en las funciones y tendrá que ser ajustada manualmente, ya que es imposible determinar a qué altura se encuentra el dispositivo usando sus sensores.

4.6.2 Métodos

- *set_height(float h_aux): void*

Esta función ajustará la altura del dispositivo a la deseada en caso de que sea necesario modificarla

- *get_FocalLength(): float*

Esta función devolverá la distancia focal de la cámara incorporada en el dispositivo, en milímetros.

- *get_VerticalViewAngle(): float*

Esta función devuelve el ángulo de apertura de visión de la cámara del dispositivo en el plano vertical, en grados.

- *get_HorizontalViewAngle(): float*

Esta función devuelve el ángulo de apertura de visión de la cámara del dispositivo en el plano horizontal, en grados.

- *get_SensorSize(): float[2]*

Devuelve un array en el que se nos indicará el tamaño del sensor de la cámara del dispositivo en sus componentes horizontal y vertical, en milímetros.

- get_Resolution(): *float[2]*

Devuelve un array en el que se indicará a que resolución está trabajando la cámara en ese instante, en píxeles.

- get_ImageCenter():*float[2]*

Devuelve un array que nos indicará el pixel central de la imagen de acuerdo con la resolución en la cual se está trabajando.

- get_Gravity(float α): *float[3]*

Devuelve un array en el cual se nos indica la gravedad la que está sometido el dispositivo descompuesta en sus 3 ejes de coordenadas, en metros por segundo al cuadrado. El parámetro α indica el grado de restricción del filtro a paso bajo que usa la función para poder aislar la gravedad. El rango oscila de 0 a 1, siendo más restrictivo según se acerque a 1.

- get_Pitch(float α): float

Esta función es una ampliación de la función ”get_Gravity(α)” en la que se usa la gravedad descompuesta para calcular el Pitch al que se encuentra el dispositivo. El parámetro α es el mismo que para la función anterior. Nos devuelve el valor de dicho ángulo en grados.

- get_Roll(float α): float

Esta función es una ampliación de la función ”get_Gravity(α)” en la que se usa la gravedad descompuesta para calcular el Roll al que se encuentra el dispositivo. El parámetro α es el mismo que para la función anterior. Nos devuelve el valor de dicho ángulo en grados.

- get_InitialDistance(int pixel_y): float[2]

Devuelve la distancia, en metros, a la cual se encuentra el objeto que estamos buscando en la imagen de nuestro dispositivo. Esta función no tiene en cuenta los posibles errores cometidos debido a las rotaciones. El parámetro pixel se lo tendremos que proporcionar a la función ya que se trata del pixel en el cual se encuentra situado el objeto de interés, variable en cada caso.

- get_EstimateDistance(int pixel_y): float[2]

Devuelve la distancia, en metros, a la cual se encuentra el objeto que estamos buscando en la imagen de nuestro dispositivo. Esta función tendrá en cuenta las rotaciones a las que está sometido el dispositivo, y se le ofrecerá al programador la opción de elegir entre cuál de los 3 métodos de cálculo explicados es el que quiere usar. El parámetro pixel se lo tendremos que proporcionar a la función ya que se trata del pixel en el cual se encuentra situado el objeto de interés, variable en cada caso.

- - get_EstimateLocation(int pixel_x, int pixel_y, int method): *float[2]*

Devuelve la distancia, en metros, a la cual se encuentra el objeto que estamos buscando en la imagen de nuestro dispositivo. Esta función tendrá en cuenta las rotaciones a las que está sometido el dispositivo, y se le ofrecerá al programador la opción de elegir entre cuál de los 3 métodos de cálculo explicados es el que quiere usar. Además nos devolverá la posición a la que se encuentra el objeto en el eje X de coordenadas, siendo el centro de la imagen el origen de éste. Esto nos permite una localización del objeto total en el plano del suelo.

Esta función tendrá en cuenta las rotaciones a las que está sometido el dispositivo, y se le ofrecerá al programador la opción de elegir entre cuál de los 3 métodos de cálculo explicados es el que quiere usar. El parámetro pixel se lo tendremos que proporcionar a la función ya que se trata del pixel en el cual se encuentra situado el objeto de interés, tanto en x como en y.

5. PRUEBAS

Las pruebas se han llevado a cabo con tres dispositivos distintos de distinta gama con la idea de comprobar si la aplicación funciona igual de bien para el rango de precios de los Smartphone o, si en cambio, los de más alta gama realizan la tarea de una mejor manera.

El primero de ellos es un Sony Xperia E, el cual es un dispositivo de gama baja.

Figura 48

El segundo es un Jiayu G4-A y se trata de un dispositivo de gama media.

Figura 49

El tercero es un Samsung Galaxy S 3, siendo este un dispositivo de gama alta.

Figura 50

Las pruebas se han llevado a cabo en una zona libre de obstáculos y con buena iluminación, ya que es necesario para que la detección funcione adecuadamente.

El dispositivo móvil se ha situado en un punto fijo a una altura de 1,30 metros, la cual es la altura aproximada a la que se situaría el dispositivo en un vehículo.

Los conos se sitúan a una distancia de 1 metro entre sí, siendo el primer cono el que marca los 2 metros. Los conos verdes marcan los metros pares y los azules/naranjas los metros impares. Por último decir que todos los conos están numerados marcando la distancia que representan.

5.1 Método 1

5.1.1 Smartphone de gama baja

Distancia real 3.10 metros

Distancia real 5 metros

Distancia real 4 metros

Distancia real 9 metros

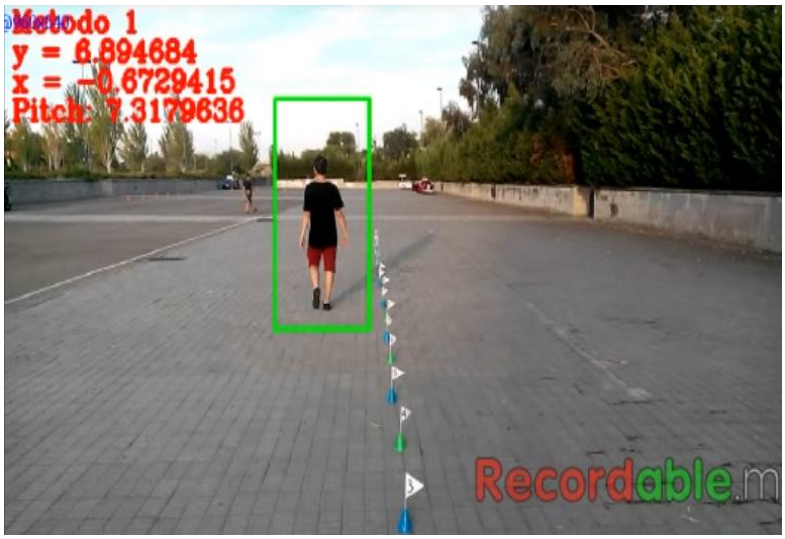
Distancia real 6 metros

Distancia real 2.50 metros

Distancia real 10 metros

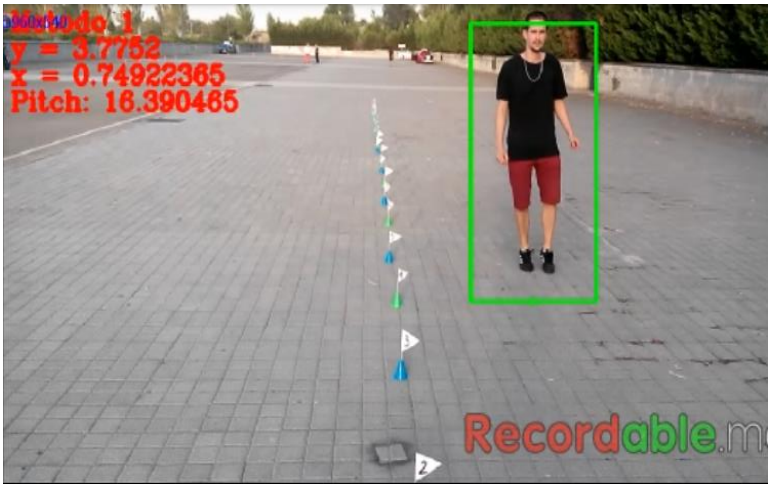
Distancia real 7 metros

5.1.2 Smartphone de gama media

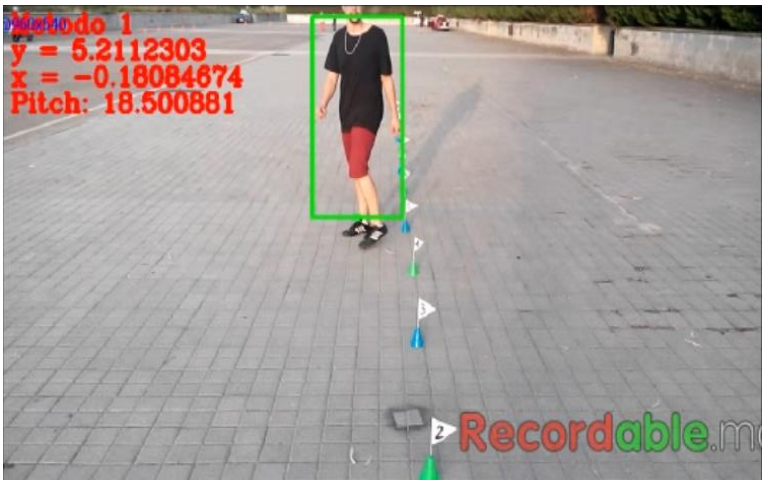


Distancia real 7 metros

Distancia real 10.50 metros

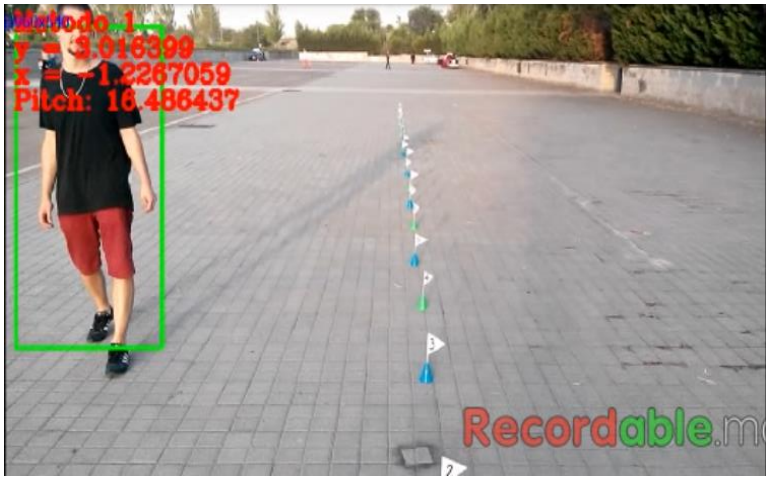


Distancia real 4 metros



Distancia real 5.20 metros

Distancia real 4.50 metros



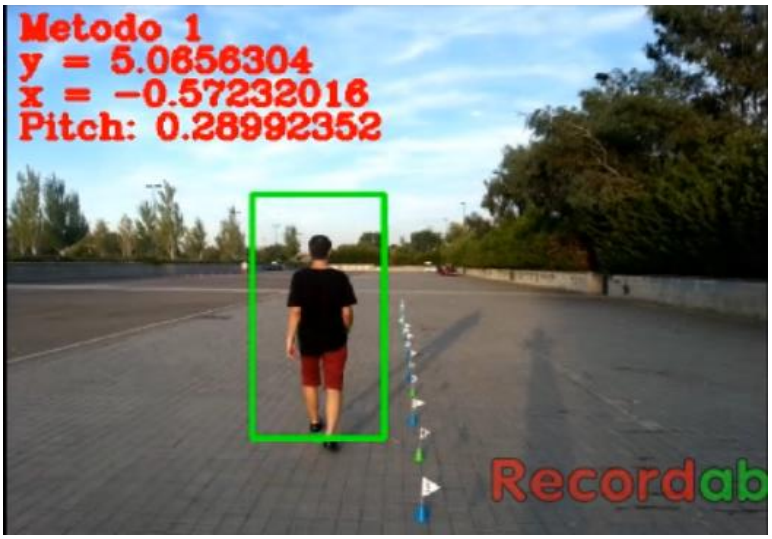
Distancia real 3.10 metros

Distancia real 8 metros

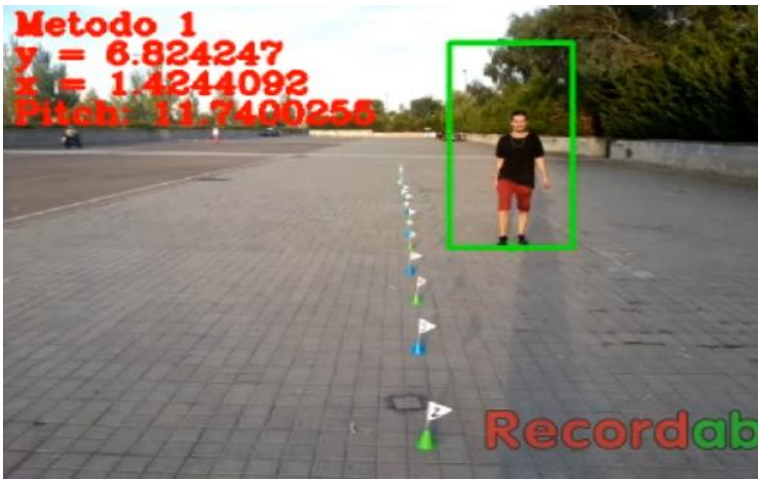
Distancia real 5 metros

Distancia real 6 metros

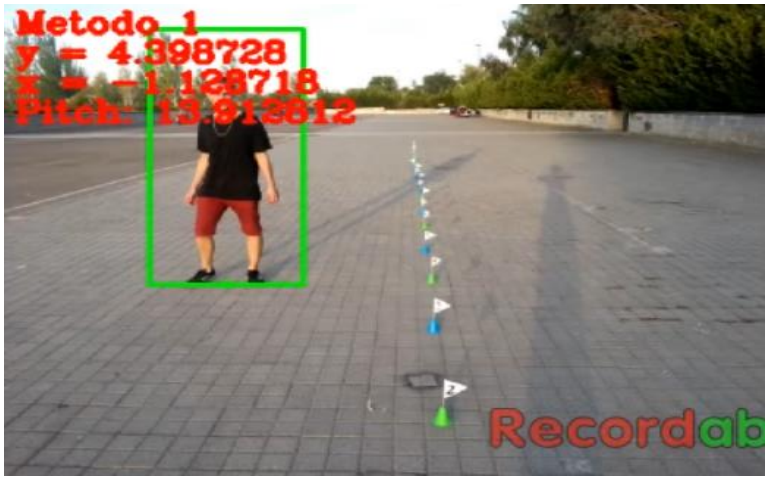
5.1.3 Smartphone de gama alta



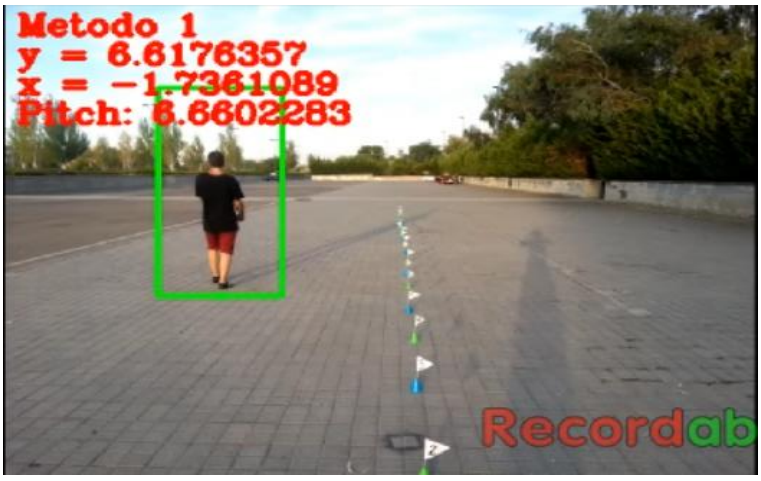
Distancia real 4.50 metros



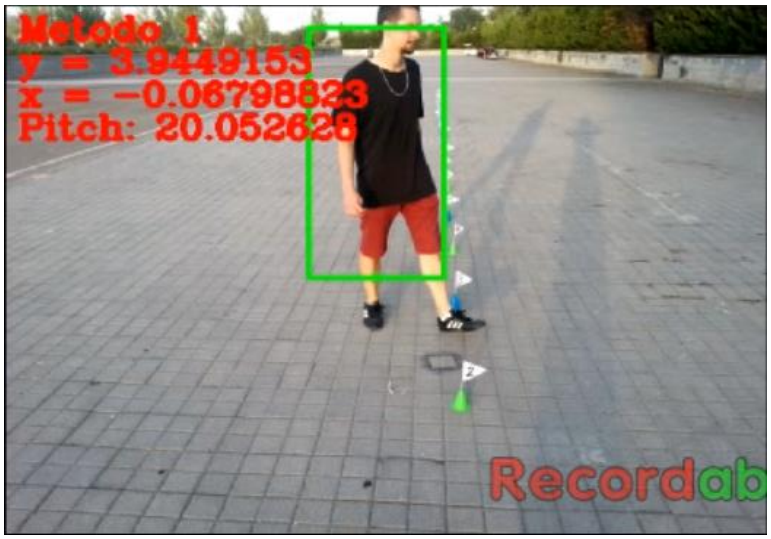
Distancia real 6 metros



Distancia real 4 metros



Distancia real 6 metros



Distancia real 3.50 metros

5.1.4 Conclusiones método 1

Según los datos observables en las imágenes anteriores se puede decir que este método funciona verdaderamente bien.

Para el dispositivo de gama media el error que se comete, con respecto a la distancia real, es prácticamente inexistente, siendo de 50 cm en los peores casos.

El dispositivo de gama alta comete un error de forma más sistemática, pero al igual que en el dispositivo de gama media, el mayor error cometido es alrededor de 50 cm.

El caso más desfavorable es el del dispositivo de gama baja. En este caso se puede observar que para distancias pequeñas, por debajo de 5 metros, el error vuelve a ser pequeño como en los casos anteriores, pero para distancias largas se comete un gran error. Esto puede ser debido a la baja resolución de la imagen en la que cada único pixel representará mayor distancia.

Como conclusión final, se puede afirmar de este método realiza una corrección del "pitch" muy adecuada, ya que calcula la distancia de manera satisfactoria sea cual sea el ángulo de inclinación al que se encuentra el dispositivo.

5.2 Método 2

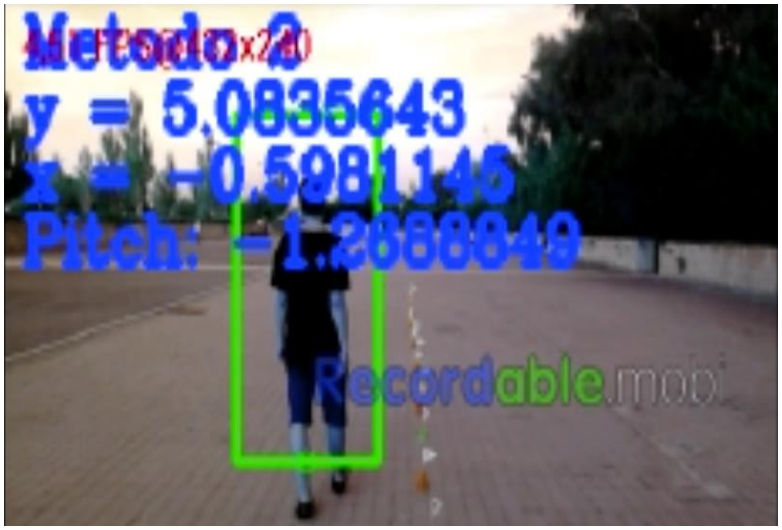
5.2.1 Smartphone de gama baja

Distancia real 6 metros

Distancia real 10 metros

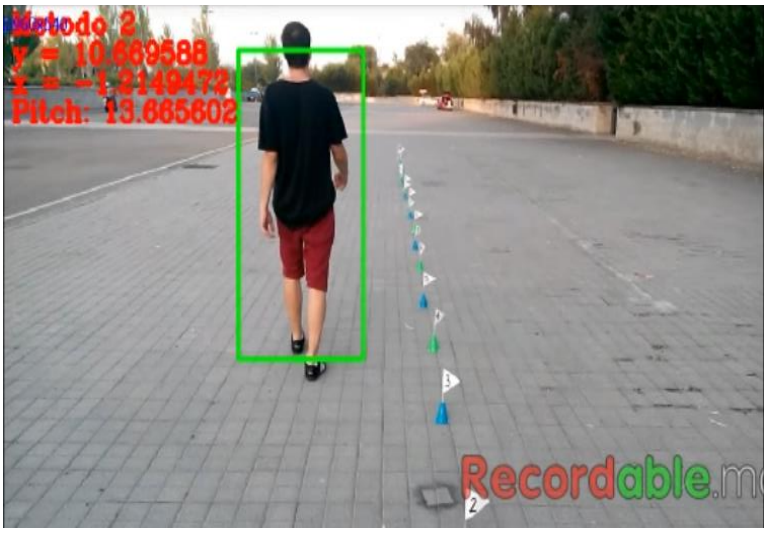
Distancia real 6 metros

Distancia real 5 metros

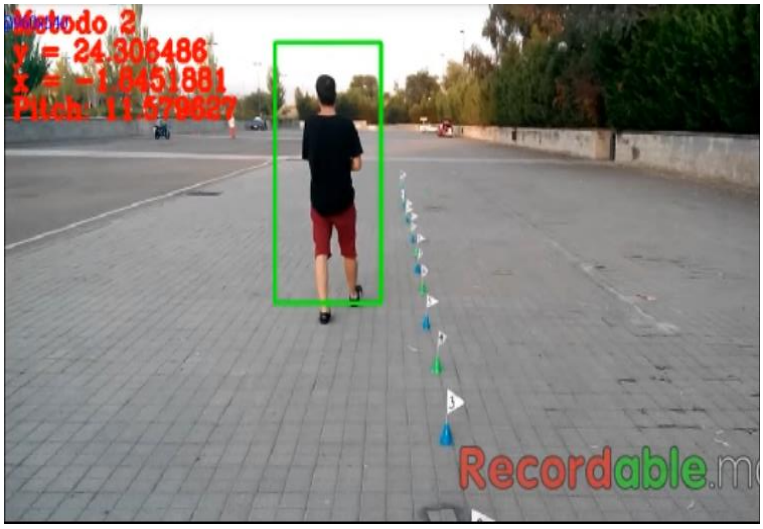


Distancia real 5.50 metros

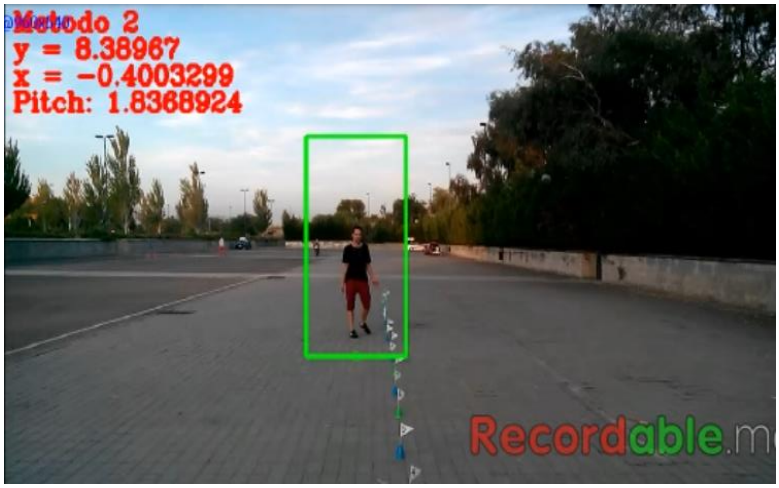
5.2.2 Smartphone de gama media



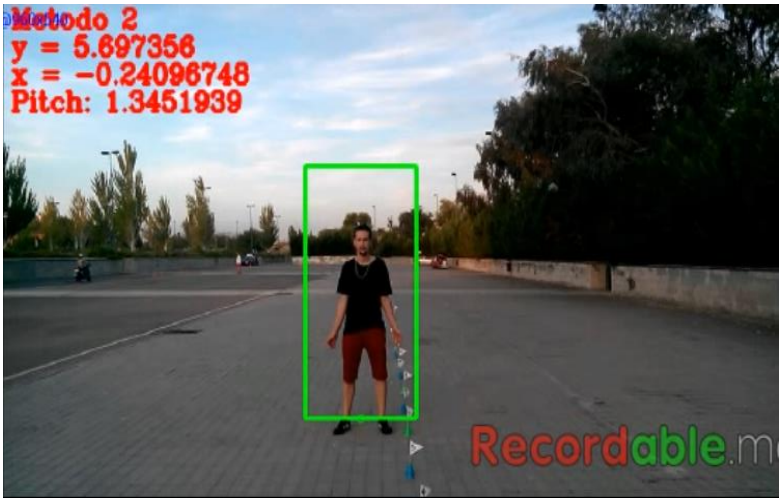
Distancia real 6 metros



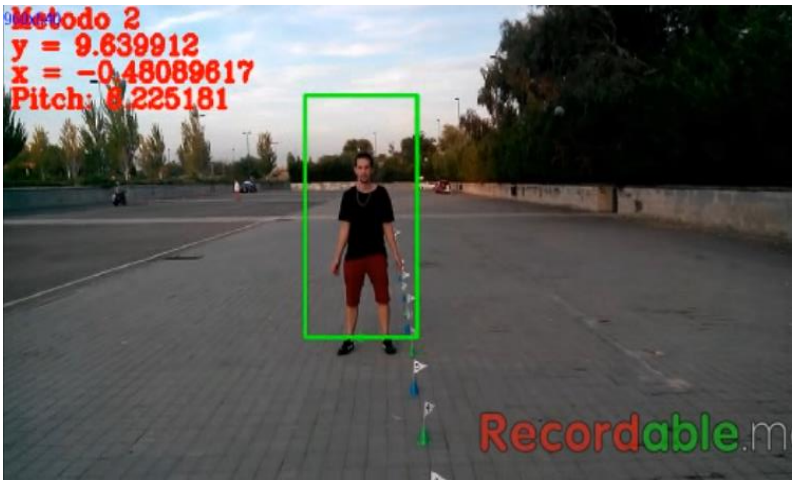
Distancia real 5.50 metros



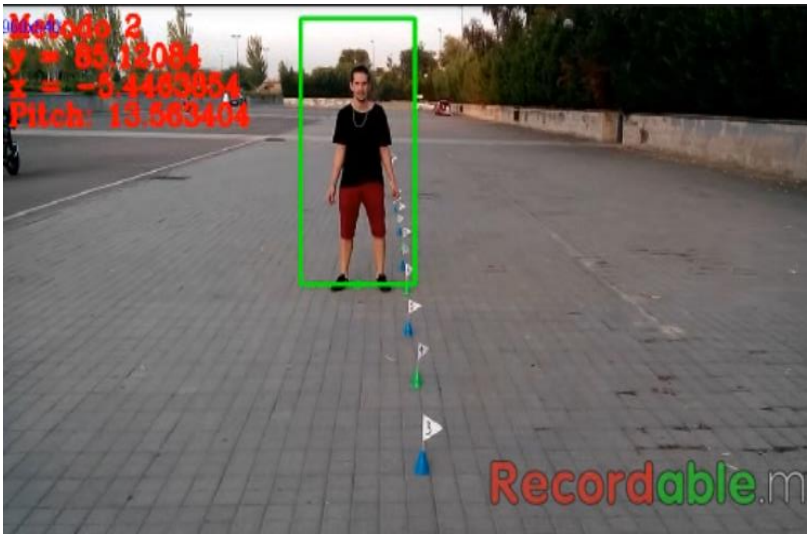
Distancia real 9 metros



Distancia real 6 metros

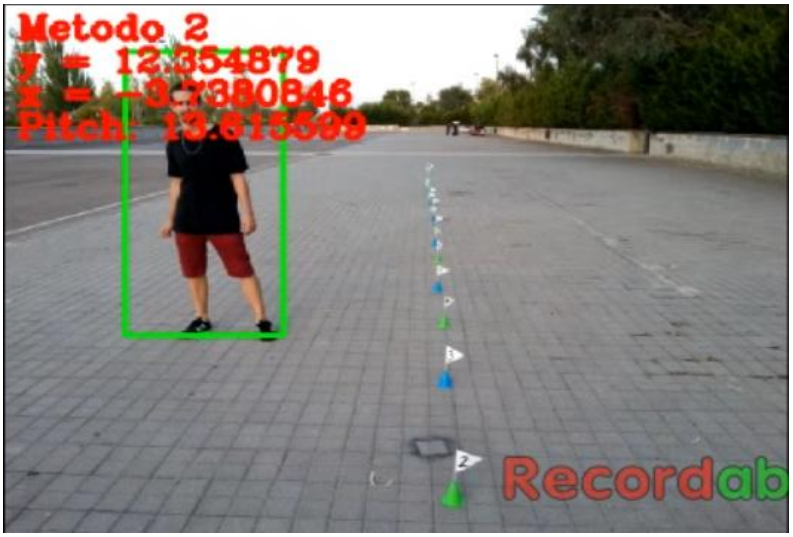


Distancia real 6.50 metros

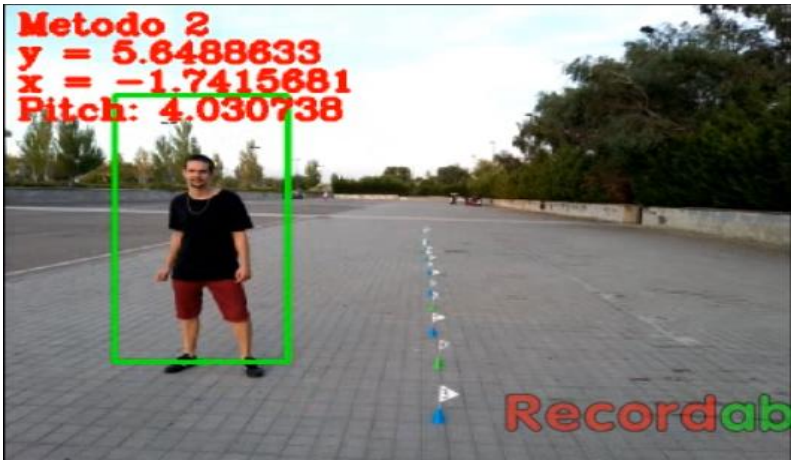


Distancia real 6 metros

5.2.3 Smartphone de gama alta

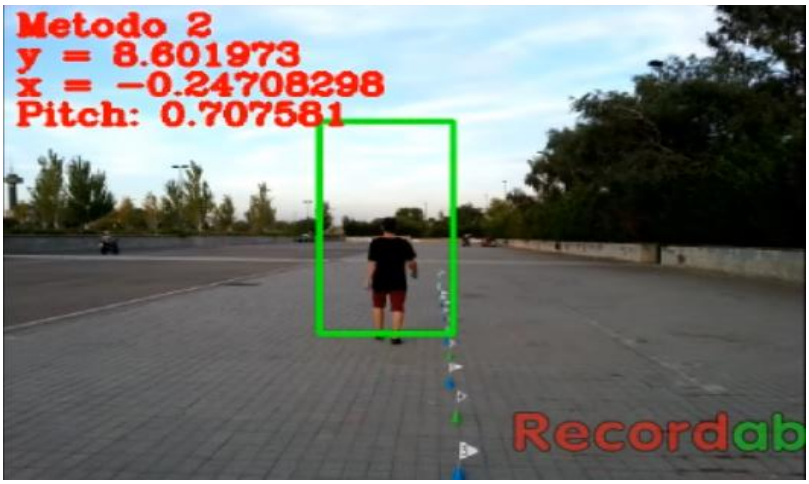


Distancia real 3.70 metros

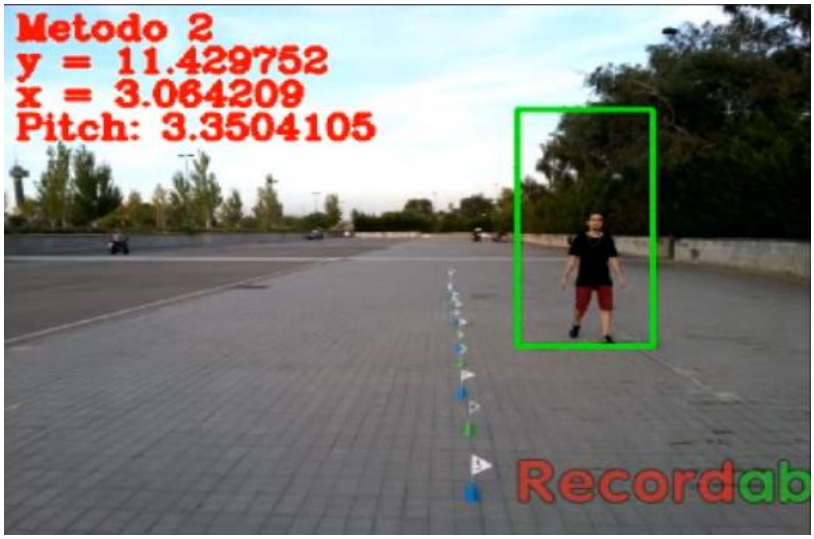


Distancia real 4 metros

Distancia real 4 metros



Distancia real 8 metros



Distancia real 7 metros

5.2.4 Conclusiones método 2

De este método se puede decir que no funciona correctamente. Si observamos las imágenes, sea el dispositivo que sea, se observa que cuando el "pitch" es considerable, el método no consigue realizar la corrección. Así por ejemplo para una distancia real de 3.7 metros y un "pitch" de 13.6 º, la distancia calculada resulta ser 12.3 metros, del todo inaceptable.

Se ve que el fallo se encuentra exclusivamente en el método de corrección, ya que si el "pitch" es despreciable, las distancias calculadas coinciden con las reales.

5.3 Método 3

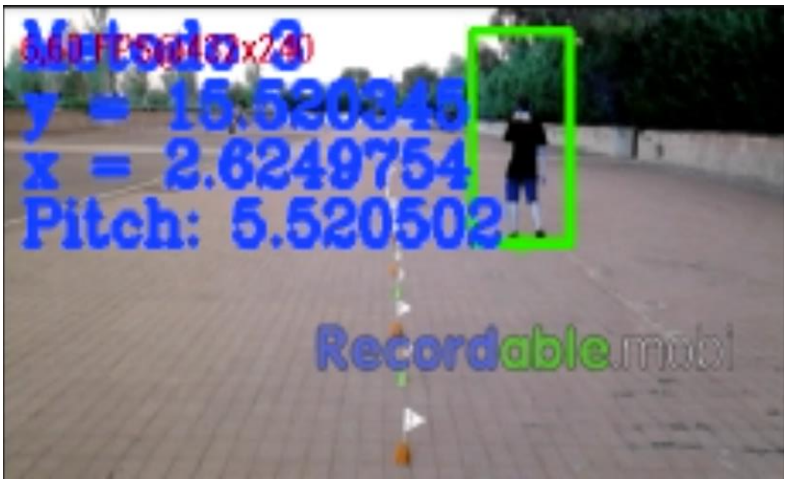
5.3.1 Smartphone de gama baja

Distancia real 5 metros

Distancia real 6 metros

Distancia real 5 metros

Distancia real 6 metros

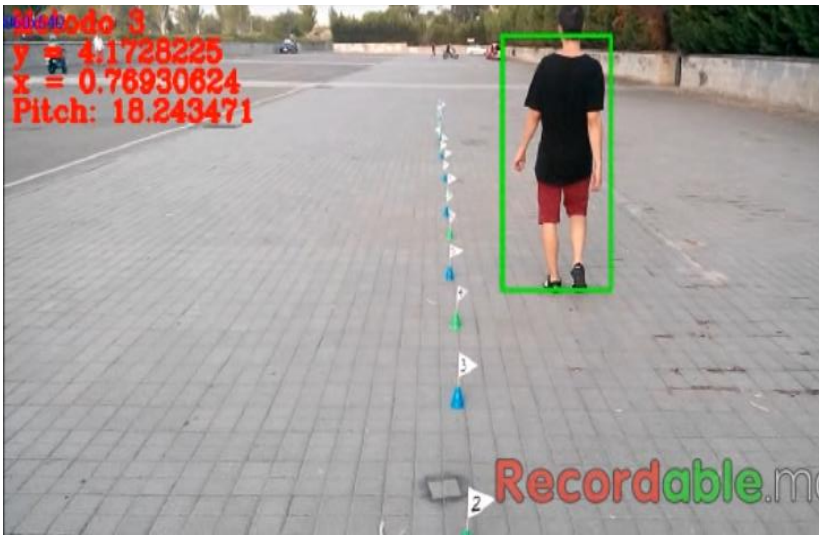


Distancia real 9 metros

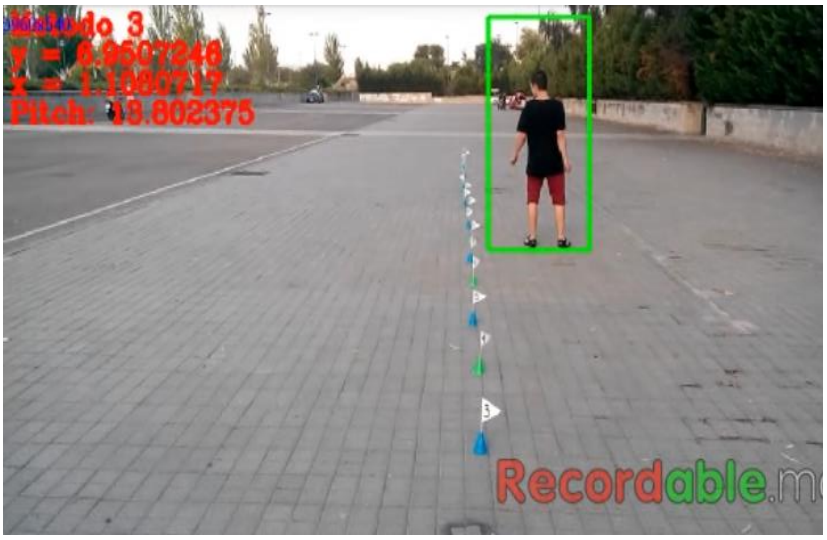
Distancia real 3.50 metros

Distancia real 5.50 metros

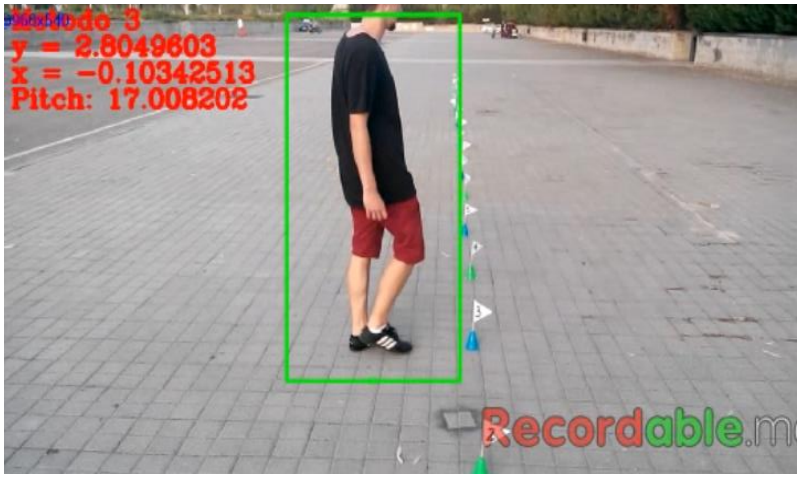
5.3.2 Smartphone de gama media



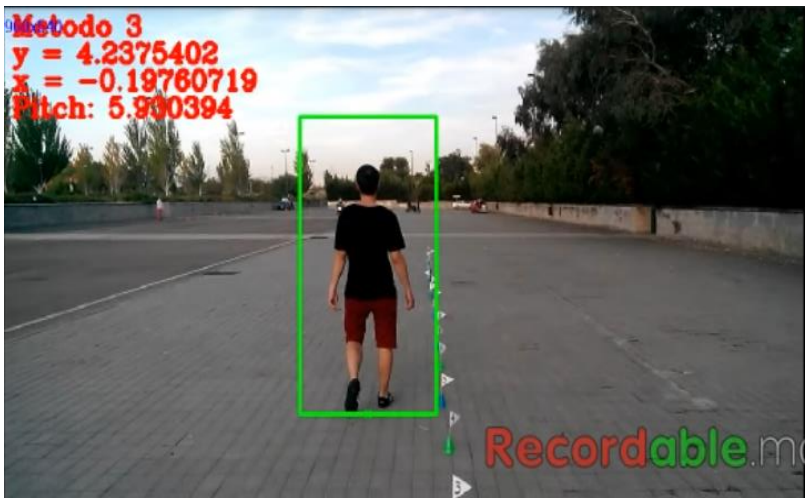
Distancia real 4.50 metros



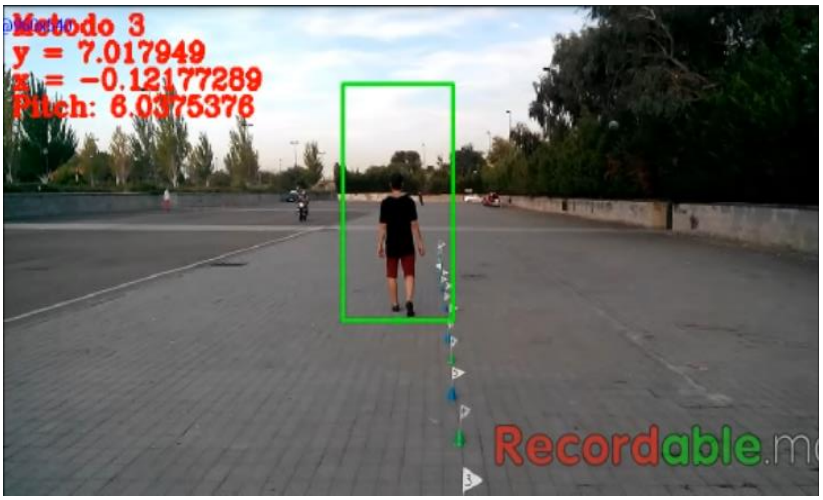
Distancia real 7 metros



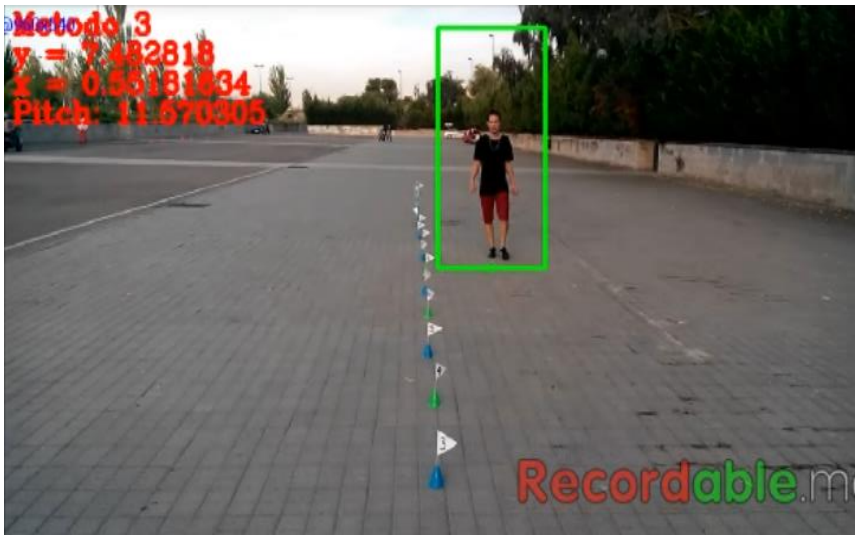
Distancia real 2.80 metros



Distancia real 4.80 metros



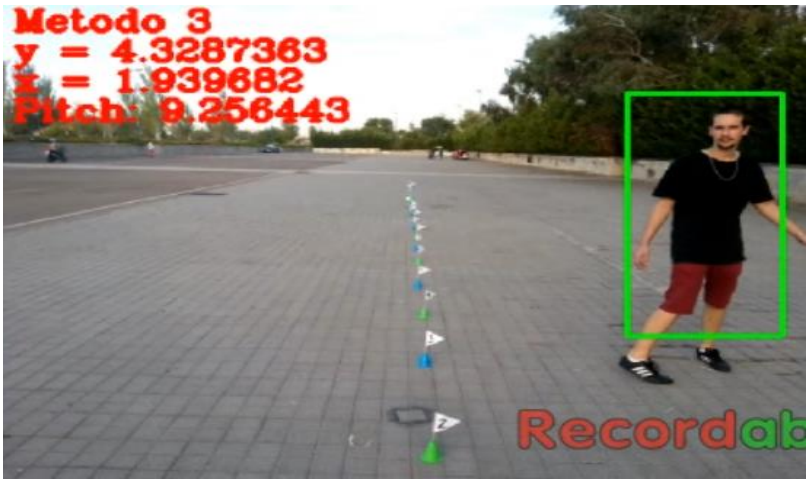
Distancia real 8 metros



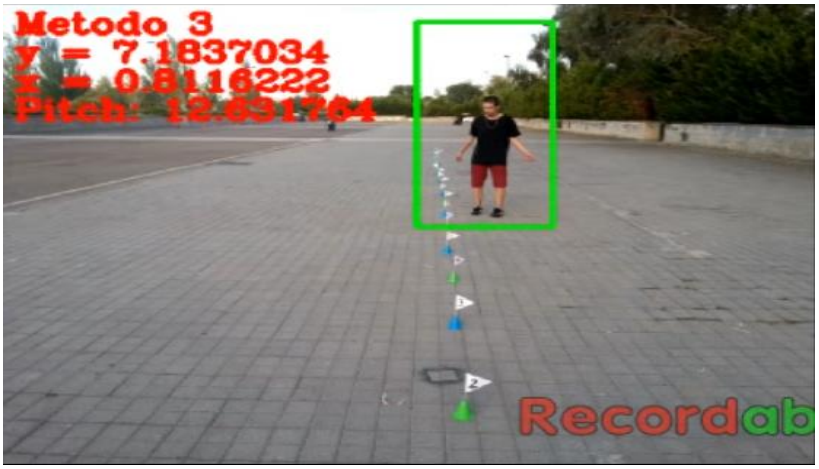
Distancia real 8 metros

5.3.3 Smartphone de gama alta

Distancia real 2.10 metros

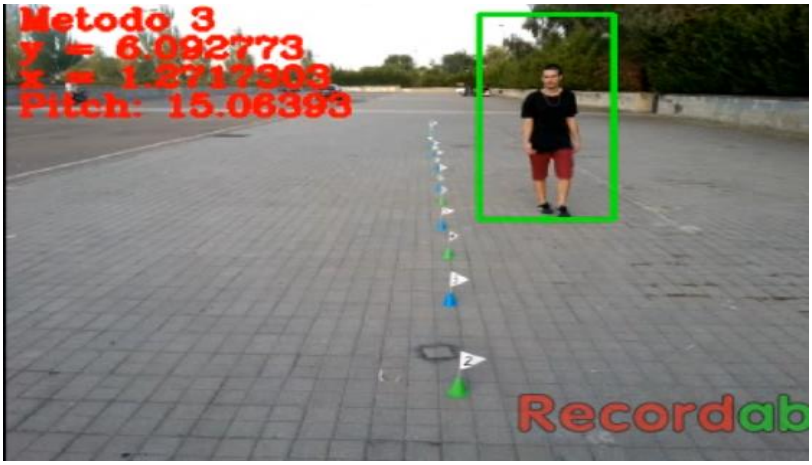


Distancia real 3.80 metros

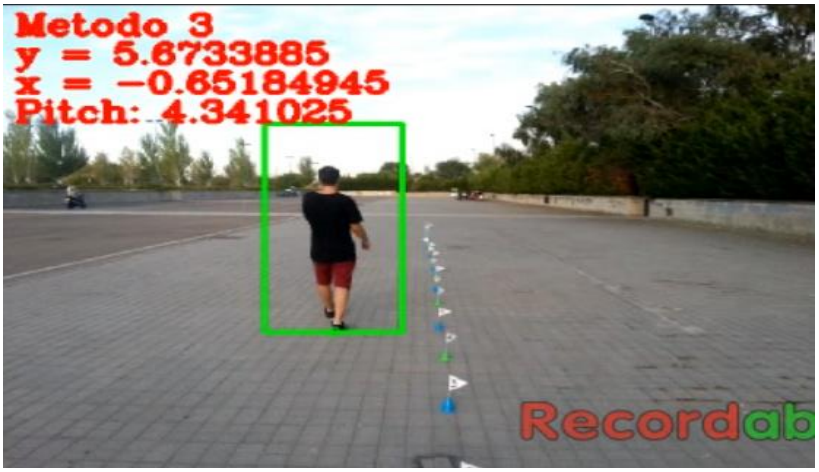


Distancia real 6 metros

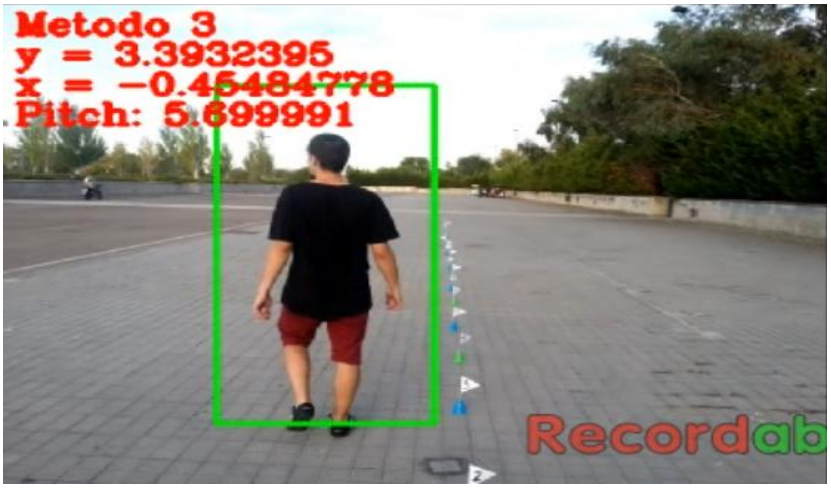
Distancia real 4.50 metros



Distancia real 6 metros



Distancia real 5 metros



Distancia real 3 metros

5.3.4 Conclusiones método 3

De este método se puede decir que funciona correctamente. Al igual que el primer método las distancias calculadas se acercan en gran medida a la distancia real a la que se encuentra el peatón.

Al igual que en el método 1 los casos más desfavorables ocurren con el dispositivo de gama baja, y sobre todo para distancias grandes.

Para los dispositivos de gama media y gama alta las distancias calculadas son bastante exactas, cometiendo un error inferior a medio metro si la distancia a la que se encuentra el peatón no es muy grande. Para distancias superiores, del orden de 8 metros o más, los errores que se cometen se encuentran alrededor del metro.

Al igual que en el método 1, se puede observar que es indiferente el grado de inclinación al que se encuentra el dispositivo ya que el método realiza una buena corrección, incluso a veces la distancia calculada es más exacta a mayor “pitch”.

6. Conclusión final y reflexión

Como conclusiones finales se puede decir que los métodos primero y tercero son muy superiores al segundo método, por ello se descarta este método como el adecuado para realizar el cálculo de distancias. No se puede escoger entre el primer y tercer método ya que con ambos se obtienen resultados muy similares, siendo ambos aceptables para realizar el cálculo.

En cuanto a los dispositivos, se puede afirmar que las funciones incluidas en la biblioteca para el cálculo de distancias funcionan para todo tipo de dispositivo. Si bien es verdad que, como era de esperar, a mayor gama de dispositivo se obtendrán mejores resultados. Esto es debido a la resolución de las imágenes tomadas por la cámara. Estas imágenes tendrán mejor resolución ya que la cámara será mejor. Esta mayor resolución se traducirá en mayor precisión a la hora de calcular la distancia, ya que lo que en realidad estamos haciendo es traducir una cantidad de píxeles a metros. Dicha resolución toma más importancia cuando queremos medir distancias más largas, ya que la precisión en estos casos es más importante. En cuanto al procesamiento de imágenes, cualquier dispositivo está capacitado para llevarlo a cabo. La principal diferencia entre gamas es la fluidez con la que percibimos la grabación de la cámara. En las gamas más bajas apenas se obtienen 5 fps (frames per second) de manera que parece que el dispositivo está continuamente bloqueándose.

Está demostrado que gracias a la tecnología se pueden llevar a cabo cantidad de proyectos e inventos capaces de hacernos la vida más sencilla. En el ámbito del automóvil existe el aliciente de salvar vidas humanas gracias a dicha tecnología. Debido a que hay vidas humanas en juego, estas tecnologías enfocadas a la ayuda en la conducción han de ser muy fiables ya que si no lo fueran podrían ser tan perjudiciales como beneficiosas. Además de la fiabilidad que se exige, existen muchas barreras tanto legales como mentales. Las legales están impidiendo que el coche de Google circule libremente en cualquier país, por ello las pruebas se están realizando únicamente en algunos estados de Estados Unidos. Las mentales se refieren a la mentalidad de los conductores de vehículos privados los cuales, en su gran mayoría, no están dispuestos a ceder su control a un ordenador, ya sea porque no se fían o por cualquier otra razón.

Debido a estas barreras, la mayoría de aplicaciones existentes que son capaces de tomar el control del vehículo están en fase de desarrollo y no se han aplicado comercialmente, aunque se prevé que el momento en que esto ocurra no es muy lejano.

7. BIBLIOGRAFÍA

[1] DGT: El número de muertos por accidente de tráfico en 2013 registra un mínimo histórico. Disponible en: http://www.dgt.es/es/prensa/notas-de-prensa/2014/20140103-balance-2013-seguridad-vial-2013.shtml

[2] Ministerio de Transportes y Comunicaciones de Perú: Seguridad vial en el mundo, diagnóstico, estadística y perspectiva en el Perú. Disponible en: http://www.presevilab2013.org/Ponencias%20PDF/EXPOSICION%201er%20CONGRESO%20IEBROAMERICANO%20SEGURIDAD%20VIAL%20OCT2013.pdf

[3] Coches: Sistemas de asistencia al conductor: ¿Qué son y cómo funcionan? Disponible en: http://www.coches.net/noticias/sistemas-ayuda-conductor

[4] CEA (Comisariado Europeo del Automóvil): Seguridad activa y pasiva del vehículo. Disponible en: http://www.cea-online.es/reportajes/seguridad.asp

[5] MotorPasion: Los sistemas de seguridad pasiva más usadas en el coche. Disponible en: http://www.motorpasion.com/espaciotoyota/los-sistemas-de-seguridad-pasiva-mas-utilizados-en-el-coche

[6] MotorPasion: Los diez sistemas de seguridad activa más conocidos. Disponible en: http://www.motorpasion.com/espaciotoyota/los-10-sistemas-de-seguridad-activa-del-coche-mas-conocidos

[7] Wikipedia: Control de velocidad. Disponible en: http://es.wikipedia.org/wiki/Control_de_velocidad#Control_de_velocidad_inteligente

[8] Gizmodo: El nuevo coche autónomo de Google no tiene volante, solo sensores. Disponible en: http://es.gizmodo.com/el-nuevo-coche-autonomo-de-google-no-tiene-volante-sol-1582478700

[9] Explicando: ¿Cómo funciona el coche sin conductor de Google? Disponible en: http://www.explicando.es/como_funciona_el_coche_sin_conductor_de_google_77

[10] Wikipedia: Android. Disponible en: http://es.wikipedia.org/wiki/Android

[11] Wikipedia: Historia de Linux. Disponible en: http://es.wikipedia.org/wiki/Historia_de_Linux#Aparici.C3.B3n_de_Linux

[12] OpenCV. Disponible en: http://opencv.org/

[13] Wikipedia: Anexo: Historia de Microsoft Windows. Disponible en:
http://es.wikipedia.org/wiki/Anexo:Historia_de_Microsoft_Windows

[14] Eroski Consumer: Siete aplicaciones para Android que hacen la conducción más segura. Disponible en: http://www.consumer.es/web/es/tecnologia/software/2013/06/12/216930.php

[15] INPENOR (Gabinete de ingeniería y peritaje del noroeste): Aplicaciones para Smartphones relacionadas con la conducción. Disponible en:
http://www.inpenor.com/2013/06/25/aplicaciones-para-smartphones-relacionadas-con-la-conduccion/

[16] ABC de Sevilla: Las diez mejores aplicaciones Android para el coche. Disponible en: http://sevilla.abc.es/mobility/las_mejores_app/android/las-mejores-app-android/las-diez-mejores-aplicaciones-android-para-el-coche/

[17] SGOliver: Curso programación Android. Disponible en: http://www.sgoliver.net/blog/?page_id=3011

[18] Vogella: Android Library Projects - Tutorial. Disponible en: http://www.vogella.com/tutorials/AndroidLibraryProjects/article.html

[19] Wikipedia: Aircraft principal axes. Disponible en: http://en.wikipedia.org/wiki/Aircraft_principal_axes#Normal_axis_.28yaw.29

[20] Wikipedia: Euler angles. Disponible en:
http://en.wikipedia.org/wiki/Euler_angles

[21] El Androide Libre: Los Smartphones como sustitutos de las cámaras digitales. Disponible en: http://www.elandroidelibre.com/2013/11/los-smartphones-como-sustituto-de-las-camaras-digitales.html

[22] Wikipedia: Pinhole camera model. Disponible en:
http://en.wikipedia.org/wiki/Pinhole_camera_model

[23] Wikipedia: Fotografía estenopeica. Disponible en:
http://es.wikipedia.org/wiki/Fotograf%C3%ADa_estenopeica

[24] Android Developers: CameraParameters. Disponible en: http://developer.android.com/reference/android/hardware/Camera.Parameters.html

[25] Android Developers: Sensor. Disponible en: http://developer.android.com/reference/android/hardware/Sensor.html

[26] Android Developers: SensorManager. Disponible en: http://developer.android.com/reference/android/hardware/SensorManager.html

[27] Android Developers: SensorEventListener. Disponible en: http://developer.android.com/reference/android/hardware/SensorEventListener.html

[28] Android Developers: SensorEvent. Disponible en: http://developer.android.com/reference/android/hardware/SensorEvent.html

[29] DE LA ESCALERA HUESO, Arturo. Visión por Computador, fundamentos y métodos. Isabel Capella; Sonia Ayerra; José A. Clares; Tini Cardoso. Pearson Educación, S.A. Madrid, 2001. ISBN: 84-205-3098-0